SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM.

1. REPORT NUMBER BBN REPORT NO. 3621	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY	5. TYPE OF REPORT & PERIOD COVERED Quarterly Progress Report 1 March 77 - 31 May 77
	6. PERFORMING ORG. REPORT NUMBER BBN Report No. 3621
7. AUTHOR(s) M. Beeler, R. Viswanathan, J. Burchfiel, J. Makhoul, J. Sussman, R. S. Nickerson, A.W.F. Huggins	8. CONTRACT OR GRANT NUMBER(s) MDA 903-75-C-0180
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency Information Processing Technology Office 1400 Wilson Boulevard, Arlington, VA 22209	12. REPORT DATE June 1977
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse; Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

packet radio, computer communications, PDP-11, TCP, station gateway, ELF BCPL, cross-radio debugging, speech compression, vocoder, linear prediction variable data rate compression, speech-quality evaluation, multidimensional scaling.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document describes progress on (1) development of a packet radio network, (2) speech compression and evaluation. Activities reported under (1) includes work on Station Software and Internetworking Research and Development; under (2) Completion of the development and evaluation of an improved automatic variable frame rate scheme for transmitting LPC data based on our perceptual modeling approach; analysis of the results of our factorial subjective speech quality tests using the multidimensional scaling procedure called INDSCAL, and interpretation of the dimensions

BBN Report No. 3621 June 1977

COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY

Part I. Packet Radio
Part II. Speech Compression and Evaluation

Quarterly Progress Report No. 10

1 March 1977 to 31 May 1977

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency or the United States Government.

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935 Contract No. MDA903-75-C-0180.

Distribution of this document is unlimited. It may be released to the Clearinghouse Department of Commerce for sale to the general public.

20. Continued

of the resulting solution space in terms of vocoder parameters.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	MEETINGS, TRIPS, PUBLICATIONS	4
III.	STATION OPERATING SYSTEM	8
	A. Modifications to ELF to Support Disk Loading and In-core Restart	8
	B. Other Enhancements to ELF	11
IV.	STATION APPLICATION PROCESSES	13
	A. Measurement Process	13
	B. Information Process	15
	C. Packet Radio Station Control	17
	D. Gateway Changes	19
	E. Improvements in Other Station Processes	22
V.	INTERNETWORKING	25
	A. Meeting at COMSAT	25
	B. TCP for the KI20	26
	C. TCP11	28
	D. Change to Internet Header, 1 April 1977	29
VI.	PR NET RESEARCH AND DEVELOPMENT ADVANCES	31
	A. PR Net Protocol Design	31
	B. Measurement of Station ROP-Processing Bandwith	32
	C. The Hop Acknowledgement Protocol	34
	D. PR Buffer Lockup	36

TABLE OF CONTENTS (CONT'D)

		Page
VII.	SUPPORT	38
	A. XNET Support	38
	B. MACN 11 Support	40
	C. Hardware Support	40

I. INTRODUCTION

The Packet Radio project relies heavily on station software for a variety of control, coordination and monitoring functions. The role of BBN in developing this software is to specify, design, implement and deliver programs which implement these functions.

The Packet Radio project had reached a point where demonstrations were confidently scheduled at the close of the previous quarter. The increasing use of the PR network at SRI, both for more frequent demonstrations and for experiments, resulted in a need for new station capabilities and improved operational ease. We have met this need during this quarter by developments in several directions.

First, the ELF operating system was enhanced by adding disk loading and in-core restart capabilities. The original cross-net loading method has proven flexible and valuable in development work and early exhibitions. The support environment at SRI exaggerated the delays inherent in cross-net loading, to the point where fast loading directly from a disk on the station PDP-11 was urgently needed. During this quarter we developed and delivered an ELF with disk loading. This is further described in section III.

Second, the need for a measurement facility has risen sharply. The PR network is now runnable for considerable periods

of time without failure, and its complexity has increased substantially. The measurement process in the station controls, coordinates and gathers results from experimental runs in the network. The results will be useful to system designers, such as us at BBN, in improving network protocols and algorithms; to system integrators, such as SRI, to detect and explain interactions and anomalous behavior in operation of an actual network; and to network analysis, such as UCLA, in comparing actual network performance against theoretical and model predictions. The basic functions of the measurement process were implemented this quarter, as detailed in section IV A. Although functioning in test arrangements set up by us, full demonstration of delivery of measurement entries is delayed until UCLA is ready to receive them. This we anticipate occurring in the next quarter. Use of the disk to temporarily store measurements is also expected next quarter.

Third, increasing use of the PR net requires availability of a directory or information service. The "INFO" process, implemented during this quarter, fills this need. The information process has been outlined in previous documentation, PRTN 182 in particular, and its implementation is described in section IV B below.

Fourth, a very urgent need has arisen for a station control module. Sometimes called "operator control" or "station terminal process" in the past, this facility allows station terminals to

be dynamically switched among the various required uses, such as labeler, INFO, XRAY debugger, gateway, and connection process. The heightened use of the SRI PR net, together with a limitation on available, physically distinct terminals in the station room, brought this need to near crisis proportions. We have designed and implemented a highly flexible module for this purpose. Called STACON, this is described further in section IV C.

Several other improvements were made to station and support software during this quarter. This work, not as major as those four areas described above, supports the major work and increases the utility of the station in various ways. These changes are covered principally in sections IV D, VI and VII.

Section V discusses progress this quarter in internetworking. PDP-11 TCP, which is used in the Packet Radio station, has been improved in various ways. Additionally, TCP for the KL20 computer, integrated into the TOPS-20 monitor, has progressed greatly during this quarter. The large amount of coding produced will, we anticipate, permit testing and initial demonstration of this service host TCP in the next quarter.

II. MEETINGS, TRIPS, PUBLICATIONS

During this quarter BBN personnel participated in several formal and informal exchanges as a part of our Packet Radio effort. A March 10 meeting at COMSAT addressed general issues of internetting, the ARPA Packet Satellite program, and TCP. BBN's attendance at this meeting was especially appropriate since we are actively working in all these areas. Further details of this meeting may be found in section V, which is devoted to internetworking efforts.

On May 2 a TCP meeting was held at ISI, which we attended. May 4 and May 27 saw meetings at Collins Radio focusing on specific Packet Radio issues. The May 4 meeting addressed point-to-point routing and the use of neighbor tables for PR net control. Through our discussions with Collins personnel at this meeting we reached agreement on the way to proceed in the next round of software implementation, affecting both station and PR unit code. Our PRTN 194, presenting our proposal for point-to-point routing, was basically accepted, as were our suggestions on maintaining neighbor tables at all times and on their contents to provide a meaningful basis for station control of the network. For the May 27 meeting, on PR diagnostics, we prepared a position statement in lieu of being able to send a representative. Issues arising from that meeting continue to be of interest to us, and we expect our internal discussions on the subject to result in a PRTN at a future date.

Packet Radio Temporary Notes issued by BBN this quarter are:

* PRTN 215/PSPWN 63, Measurement File Delivery Specification. This documents the design of the delivery mechanism for measurement files. At the request of Packet Satellite Project members this PRTN, first released last quarter, was also issued as a Packet Satellite Project Working Note. It bears on means the PS project may find useful for delivering their measurement results.

* PRTN 216/PSPWN 62, Specification for an ELF System with Disk and Network Loading Facilities. This document provided the framework for implementation of the disk loading ELF it describes, which was also accomplished this quarter. The need for disk loading has been touched upon in section I above, and further details of this facility are given in section III A below. The documentation and coding of this represents a major improvement in the utility of the ELF operating system. This PRTN was also issued as a PSPWN at the request of the Packet Satellite group, since ELF is used in the gateway PDP-11 computers associated with that research.

* PRTN 217, Measurements of Station ROP-Processing Capacity. At the close of last quarter, at a meeting at SRI, the issue of Repeater-On-Packet (ROP) processing capacity arose. The station examines ROPs from PR units in an operating network

to deduce network connectivity. Concern was expressed that the station might be flooded with ROPs faster than it can process them. To address this concern, we designed and executed measurements of the station's ability to process ROPs. This PRTN reports the results of these experiments, which agree with the contention that conditions of very high ROP production could exceed the station's processing capacity, but that this would not be expected very often. These experiments were performed with an early version of the measurement process, which gave support and encouragement to our implementation efforts.

* PRTN 218, <u>Station Control Module Specification.</u>
  This PRTN documents the station control process, STACON, explaining its various modes and features. The orientation is principally that of an operator manual, rather than that of a technical design. Thus, this PRTN will become a part of the Station Operator's Guide which accompanies delivery of the next version of station software.

* PRTN 223, <u>Alternate Routing Reconsidered.</u>
  The design of new routing protocols in the PR net has often been influenced by the tacit assumption that some form of alternate routing is not only desirable but even necessary. PRTN 223 attacks this notion, examining in turn various aspects of alternate routing. The cost and potential benefit of each aspect is discussed, and the conclusion is reached

that alternate routing is not absolutely necessary. Indeed, in some instances of network design, alternate routing may not be cost effective.

In addition to the formal literature and meetings outlined above, we participated in several fruitful informal negotiations this quarter. We reached agreement with Collins Radio personnel on solutions to Channel Access Protocol (CAP) problems, especially hop acknowledgement design. The subject is how one PR unit properly identifies a packet it receives, as a hop acknowledgement.

We have been in frequent contact with UCLA regarding changes and additions to the measurement file entry specification (PRTN 212). This activity will help smooth start-up of measurement processing at UCLA, since they are aware of the evolution of the format of entries.

We have corresponded with Collins Radio on PRTN 194 (Point-to-Point Routing Proposal), and with Network Analysis Corporation about PRTN 192 (Route Assignment Proposal). These interchanges help to fine tune the PR net designs and keep the members of the PR working group involved and informed.

III. STATION OPERATING SYSTEM

Reliable, efficient operation of the packet radio station relies on an operating system which provides an adequate environment in which the applications processes may run. The ELF system continues to serve this purpose, and our efforts to keep it current with the growing needs of the packet radio project are described in this section.

A. Modifications to ELF to Support Disk Loading and In-core Restart

As work has progressed on the packet radio station and gateways, it became apparent that an ELF system was needed which would be capable of being loaded from the disk or network and which could be restarted in core. The ELF system is used in several different environments. For example, in an operational environment, it is desirable to be able to load the ELF and all user processes from disk and to restart this system by simply setting the proper bootstrap address in the switches and pushing the start switch. On the other hand, in a debugging environment, it is desirable to keep several versions of the ELF and user programs on disk, to be able to selectively load some programs from disk and others from across the net, and to be able to individually start and halt processes. Still another environment is in the internetting community, where it is helpful to have the facilities of cross-net debugging to remote gateways, but where the bandwidth to control gateways must be minimized because of the low bandwidth lines available.

With these constraints in mind, we set the following specific goals in designing our modifications to the ELF.

1. It must be possible to load and start an entire system (ELF and user programs) without using cross-net facilities.

2. All debugging facilities could be provided cross-net (i.e., no debugging facilities would have to be provided local to the PDP-11).

3. It must be possible to load part of the system from PDP-11 disk and part of the system cross-net.

4. It must be possible to retrieve and store core images on PDP-11 disk using the cross-net debugger; it should not be necessary to have access to a terminal locally connected to the station to access the disk.

Design and implementation of the modifications to the ELF were completed in this quarter. This work included:

1. Writing a bootstrap program which could boot an ELF from disk or from across the network.

2. Adding new commands and modifying existing commands in both XNET, the module which runs on a TENEX or TOPS-20 system, and communicates with the PDP-11, and in the debugger in the PDP-11.

3. Modifying the code used to establish the 11BCPL runtime environment and modifying the 11BCPL library.

4. Making modifications to the ELF kernel, particularly in the initialization code and in the virtual storage allocation procedures.

The primary features of the new ELF system are:

1. The ELF and user address spaces can be stored on and retrieved from disk. In particular, an operational Packet Radio station or Satellite Net gateway can be reloaded and restarted by setting the bootstrap address in the PDP-11 switches and restarting the machine.

2. The ELF and user processes are in-core restartable. It is possible to specify which user processes are to be restarted and which user address spaces are to be preserved when the system is restarted.

The ELF with the new disk loading facilities was delivered to SRI for use in the Packet Radio Station, to University College London for use in the Satellite Net gateway and to Lincoln Labs for use in speech experiments on the Satellite Net.

B. Other Enhancements to ELF

Several additional modifications were made to the ELF system this quarter. These modifications were in three areas: addition of a system primitive for process naming, changes to the XNCP to process RFNMs, and addition of timestamping code to the XNCP and VDH reliable transmission protocol code.

A system primitive was added to the ELF kernel to provide a process naming facility. This primitive allows a process to declare a name to be associated with its process ID number. Conversely, the primitive can be called with a process ID number, and will return the associated process name. This primitive has been used in all the P.R. net station and Satellite gateway processes. This facility is also used in the cross-net debugger. When an address space is debugged, the debugger uses the primitive to obtain the name of the top level process in that address space and returns the name to the user.

Changes have been made to the XNCP (experimental network control program) to process RFNMs. The XNCP had previously sent messages to the network without waiting for the RFNM for the previous message. It was suspected that this may have been flooding the network in some circumstances. The XNCP has now been modified to account for the number of messages outstanding on the network, i.e., the number of messages sent minus the number of RFNMs received, and to discard messages when the number of messages outstanding is above a given number.

Code was added to implement timestamping of packets received or sent on a network connection to the ELF. The timestamps have been implemented as an internet option for use in the Satellite Net as explained in PSPWN 52. Briefly, each timestamped packet includes a specification of which level at which to timestamp the packet and a set of timestamps. Packets may be timestamped at several levels: in the process which is the source or destination for the packet, in the gateways when a packet is received or sent to the network, in the SIMPs, when a packet is received or sent to the gateway, and in the SIMPs, when a packet is received or sent on the Satellite channel. In the gateway, code to timestamp packets was implemented in the IMP11A driver which sends and receives packets from the ARPANET and in the VDH reliable transmission module which sends and receives packets from the Satellite Net.

IV. STATION APPLICATION PROCESSES

Applications processes implement the various tasks which the packet radio station must perform. Efforts during this quarter have considerably expanded and refined the repertoire of functions available in station software. This section describes the need for this increased capability and the means by which it is now available.

A. Measurement Process

Two major features of the measurement process were implemented this quarter -- control and collection of cumulative statistics (cumstats) from PRs, and transfer of measurement data from the station using TCP.

The mechanisms for controlling and collecting PR cumstats had already been designed, as reported last quarter. This quarter we received from Collins the PR software to implement measurements, and documentation of the PR measurement parameters and packets. Commands for specifying PR parameters (cumstat interval, whether or not to report neighbor tables and buffer snapshots, and snapshot interval) were added to the measurement process. To initiate measurements from a PR, the measurement process sends it a control packet specifying the parameters. The PR replies with a control data packet confirming that measurements have been initiated and at what time (by the PR's internal clock). In exercising this PR control at BBN, we found

a number of bugs in the cumstat control packet (which contains code to be executed by the PR) and in the PR code to generate cumstat packets. The latest software delivery from Collins fixed these bugs and works well with the measurement process.

A TCP program is now available in the station (see section V). The measurement process uses TCP to send measurement data, as it is received or generated, directly to a remote server for filing and later analysis. In order to test this TCP transfer, we wrote a data receiver which resides in the station and simply prints all data it receives. This receiver will not be part of the station; the data will actually be transferred to UCLA. We are negotiating with UCLA to accomplish TCP transfer to their machine.

In addition, the implementation of connection process cumstats was completed.

As the implementation has progressed, changes and additions to previously documented measurement file entries have been made. Documentation of these changes will be published in a revision of PRTN 212, "Specification of Measurement File Entries," when the measurement process is delivered to SRI. Meanwhile, we are keeping UCLA posted on all changes so that their analysis programs will not fall behind.

B. Information Process

A packet radio network user may terminate his network activity and later reestablish connection with the network on a different terminal. Between these times the user is unreachable from the network. Another user desiring to contact him needs to find out what his status is: unreachable or, if reachable, via what path. Hence, knowledge of the mapping between user name and device ID at which that user is present is needed. The Information Process was designed to meet this need.

This quarter the Information Process (INFO) was written and added to the station. It keeps track of user name/device correspondence by accepting packets from devices informing it when a user starts or finishes using a device. Devices can ask the information process where a given user is, or ask for the entire list of name/device correspondence. In addition, the information process allows a user to converse with the station operator.

The user name/device ID table maintained by INFO can be updated by the operator, if necessary (for instance in the case of a device crashing). Also, the station operator might want to find out where the net users are. Therefore, all the commands available from devices are available by typing directly to INFO (except for the command that allows a user to converse with the station operator since the operator does not need to talk to himself via a keyboard).

Although the way a user uses the information process depends on how the TIU is implemented, it will look roughly as follows. The TIU will recognize a command meant for the information process and send an appropriate packet, printing the reply. What the user might type is indicated in lower case, with INFO's answer indicated in upper case. Editorial comments are in parentheses.

```
login johnson
JOHNSON LOGGED IN ON  4251 (where 4251 is the device ID in
hexadecimal)

(and after doing some work, Johnson might decide to send a
message to Smith and Jones, so he will want to know where to
reach them)

where smith
SMITH LOGGED IN ON  5521
where jones
JONES NOT LOGGED IN

(there are a few other people who might serve as well as Jones
for Johnson's problem, so he decides to find out everyone he can
reach)

all
         NAME
3562  AARDVARK
6731  RHUBARB
8211  ARTICHOKE
5521  SMITH
6279  PETUNIA
4251  JOHNSON

(well, nobody who can answer the problem is logged in, so Johnson
decides to ask the station operator the question)

help Where is the best Chinese restaurant in Boise, Idaho?
I'm going to a conference there and don't want to starve.

(eventually the station operator answers)
THERE ARE NO CHINESE RESTAURANTS THERE.

well what can I do then?
```

```
I SUGGEST YOU SKIP THE CONFERENCE.  BYE.
CLOSING LINK
```

(having obtained that information, Johnson is finished with his work for the day, so he decides to leave)

```
logout
JOHNSON ON  4251 LOGGED OUT
```

Since most of the function of the information process centers around processing received packets, and there was no available source of such packets, we had to devise a means for a test program to send it packets. This was accomplished by patching the station PR (after making sure the label process had a chance to label it) to send any packets it received back to the station. A test program, another process in the station, was then written which sent packets to the station PR.

C. Packet Radio Station Control

The design, implementation, and debugging of the station control module of the packet radio station (STACON) was completed during this quarter. STACON serves as a switching matrix between processes running in the station and terminals which are either attached to the station or any TENEX or TOPS20 using XNET. The principal motivation for STACON is that it is uneconomical to supply a separate terminal for each station process. STACON, however, also enhances station operation in other ways. For example, more than one terminal can receive the output from a given station process.

STACON is written in machine language and occupies slightly over 4096 words of PDP-11 code and table space. It runs as a single ELF process which is multiplexed among the various tasks of handling the input and output of each terminal and process. A station operator at each terminal is presented with a command language which permits him to exercise the various features of STACON. These include connecting and disconnecting from station processes, assigning or deassigning station processes, setting values of station parameters, and producing printouts of CPU time consumed by all the station processes, as well as other functions. The command language is patterned after that of the TENEX EXEC. Commands may be abbreviated and completed. On-line help is available using question mark.

One command allows a terminal's keyboard to be used to type in to a station process. Thus a station process can be written as if it had a private terminal and then given commands and information from any station terminal via STACON. More than one terminal may be inputting to a process simultaneously, thus providing a capability much like the TENEX ADVISE command. This would be useful in cases where local personnel are being assisted from remote locations.

The output stream from any process to any terminal may be handled in a number of different ways. It may be turned off entirely by disconnecting the terminal from the process. It may be buffered until the terminal is idle and then the terminal

switched to that process's stream for printing. It may be blocked until the terminal is specifically directed to that stream. Finally, it may be discarded if the buffering capacity is exceeded.

The output stream from a process may contain error messages and warning messages which are delimited in the stream by special characters. Such messages are not discarded in discard mode and temporarily elevate the priority of that stream such that warning messages will be printed in preference to any ordinary output. Error messages override any other output except other error messages. In all cases, when a terminal switches from one output stream to another, that fact is explicitly identified on the terminal.

D. Gateway Changes

The design of the gateway was changed to accommodate TCP11, to improve the modularity of the program and to streamline the process flow for greater efficiency. Basically, each source of packets is managed by a separate process. E.g., the TCP is a process, a process receives packets from the PR net, and a process receives packets from the ARPANET. The gateway provides a way for a source of packets to send them to the "proper place". To the source process, the action is nearly identical with that of writing the packet onto some IO device. The source process retains ownership of the packet buffer and can manage its storage in whatever way it sees fit. When the packet has been delivered,

the source process receives a signal just as if an IO transfer had completed (indeed, in some cases, an IO transfer did complete). The only additional action taken by the source process that would not be done in the case of an IO transfer is to call a routine in the central gateway code which performs any actions required by the send routine which transmitted the packet (see below).

A central gateway routine is responsible for routing the packet. A list of "connections" is maintained which gives each of the addresses to which a packet may be sent. These addresses have three degrees of specificity. They may be just a net number, a net and host number, or a net, host and format. Thus the TCP sets up a connection for packets addressed to whatever net and host the TCP wants to call itself and TCP format. The ARPANET routines set up one connection for net = 12. The packet radio net routines using SPP set up specific connections to each active destination as well as a listening connection for all inactive destinations.

When a packet source wants to send a packet, it calls a central gateway routine which searches the list and selects the most specific connection which matches the packet's destination. If no such connection is found, then the routing tables are consulted to find a gateway to which packets addressed to the specified net may be sent. If such a gateway is found, then the connection list is searched a second time to find the connection

to use to send to that gateway. If no connection is found by either of these methods, the transfer is marked as having completed in error and a signal is sent to the source process (note the source process is running, so it signals itself). If a connection is found, then the entry for that connection contains the address of a routine which is called to send the packet.

The routine to send a packet may do anything necessary to send the packet, but must give the appearance to the source of having done an IO transfer. For example, the ARPANET send routine simply starts an interprocess port IO transfer to the XNCP. The packet radio send routine, however must coordinate sending packets with managing an SPP or non-SPP connection so, in this case, the send routine signals the process managing the PR net connections specifying the packet needing to be transmitted and then returns. When that process is scheduled, it will send the packet and after transmission is completed, signal the source process of that fact.

In order to avoid extraneous process switches, it is desirable to allow the source process executing in the send routine for a particular network to initiate an output transfer itself. This means, however, that the source process will directly receive the IO completion signal when the transfer completes. Since, in some cases, it is necessary for the send routines to maintain information about such transmissions, the source process is required to call a central gateway routine to

provide the opportunity for the send routines to account for the completion of the transfer. Currently, that action consists solely of decrementing a "writes in progress" counter associated with the connection, but the extension to perform more involved procedures will be trivial.

The initialization of the gateway is purely table driven. This means that the central gateway routines are unaware of the precise configuration of the gateway. There is one file which specifies two tables which define the configuration of the gateway and the local net addresses of the gateway. These tables also currently define the addresses of other gateways and thus the routing of packets. This aspect will change when dynamic gateway routing is implemented. These tables also specify the top level routine of each network specific process and thus provides the basis for creating a process to handle each network to which the gateway is attached.

E. Improvements in Other Station Processes

New versions of the connection and control (label) processes were delivered to SRI this quarter in response to changing network protocols and needs. First, a change to the hop acknowledgement protocol (for reasons discussed in section VI) made it no longer legitimate to use hierarchy levels 0 and 15. The control process was thus changed to not assign these levels; the station PR is now placed at level 1. Second, Collins implemented distress ROPs (DROPs) this quarter, so that PRs could

V. INTERNETWORKING

An important part of our work involves a viewpoint in which the packet radio network is but a component of a larger internet or "catenet." Issues involved in design and implementation of protocols to accomplish internet communications are addressed in the paragraphs below.

A. Meeting at COMSAT

Virginia Strazisar attended the Eighth Packet Satellite Program Working Group Meeting held at COMSAT in Washington, D.C., March 10-11. At this meeting, it was decided to install a 50 KB line from the Etam SIMP to the BBN gateway, rather than to move the gateway to the Washington area. Availability of the gateway at BBN has helped greatly in checking out gateway code for the ARPANET -- Satellite Net gateways. The need for an internetting monitor and control center was discussed. The participants agreed that it is important to develop tools for monitoring and debugging in the internetting environment. Various internetting problems such as flow control and routing were also discussed.

Following the Satellite project meeting, Bill Plummer, Jerry Burchfiel, and Virginia Strazisar attended the TCP meeting, which was also held at COMSAT. At this meeting, Vint Cerf distributed the TCP version 2 specification. Bill Plummer discussed some of the problems with the current TCP protocol and suggested solutions.

B. TCP for the KL20

The Internet host-host Transmission Control Protocol has reached a point of temporary stability. Version 2 of the protocol is in everyday service at the SRI-KA TENEX site and is also placed in operation at ISIC TENEX during Packet Radio demonstrations. Since the BBN Packet Radio group is now using BBNA, which is a TOPS20 system, TCP cannot be run at BBN. This is because the TOPS20 monitor does not support the JSYS trap mechanism or the special JSYSs installed specifically for TCP.

Given the above consideration and the lackluster performance of the TENEX TCP, it was decided to explore the "best, most efficient" implementation of TCP. The major areas where improvement was sought are:

1. Running on the KL20/50 would automatically produce a speed increase of two to four simply due to the faster processor.

2. Handcoding the TCP, rather than using the BCPL version which runs on TENEX.

3. Locating the TCP within the TOPS20 monitor, eliminating the need and overhead associated with JSYS traps. The TENEX version runs as a user process.

4. Interfacing the TCP to the IMP driver interrupt level code through a separate set of queues parallel to those used for the ARPANET NCP (Network Control Program).

5. Arranging the IMP driver output code to automatically use 32-bit mode for TCP messages. Similarly, 32-bit input mode is selected as soon as enough of the header of an incoming message has been read to know that it is intended for TCP.

6. Packetizing (and reassembling) directly from (into) the calling user's address space. This means that the data portion of a packet sent (received) is copied only once.

7. Proposing changes to the basic protocol, which if accepted by the TCP working group for Version 3, would further increase the efficiency. The two main areas of concern have to do with flow control when end-of-letter marks are sent, and attempting to make user buffer addresses be related to packet sequence numbers by a simple, relatively constant, relation.

Coding of the Transmission Control Program (TCP) to run in the TOPS20 monitor has been completed. A substantial portion of the code was checked out in user mode by substituting dummy routines for the missing monitor routines. Following this initial debugging, the code was integrated with the monitor sources and a private, single-pack system constructed for debugging purposes. Debugging in a live monitor continues.

Initial indications are that the code is roughly a factor of five smaller in the assembly language version than the TENEX BCPL

version. It is hoped that the program will see a comparable decrease in execution speed. The TOPS20 version has retained the extensive internal metering which is available in the TENEX version so that quantities such as CPU consumption per process and per packet will be available once the debugging has finished.

A special effort has been made while coding to maintain backwards compatibility with the TENEX operating system so that TCP may be retrofitted into the several TENEX systems where it will be required. Doing this will also permit accurate evaluation of the advantages of handcoding, etc. of a major system component on a particular processor. Further comparisons may also be made between the same program (i.e., TCP) running in the best situation on a TENEX and on TOPS20.

C. TCP11

The version of TCP written by Jim Mathis of SRI (TCP11) was adapted to run in the ELF environment of the packet radio station to be used principally for delivering measurement data to a host for processing and analysis. The changes necessary were principally confined to the interfaces between the user and TCP11 and TCP11 and the network. As originally written, the user interface used simple subroutines since the environment was not a virtual memory one. This was changed to use user EMTs as the calling mechanism and to pass data by manipulating the address space of the TCP program to provide a window into the user's space.

There were a number of instances of network specific actions being taken in the mainstream of the TCP code. Recommendations were made in order to clean up the interface between the TCP and the network so that it would be suitable for both the environment of a packet radio terminal and that of the packet radio station.

The startup procedure and process structure were also modified since, in the MOS environment, all processes are pre-defined whereas, in the ELF environment, processes are created dynamically. It was also necessary to resolve differences in system calls between the two operating systems.

When the changes were debugged, they were incorporated into the primary version of TCP11 by SRI personnel so that subsequent updates have automatically been incorporated in both the ELF version and the MOS version.

D. Change to Internet Header, 1 April 1977

Prior to 1 April space for a 32-bit timestamp was included in every Internet message header. Conceptually this was part of the (extended) local header since it was not included in the checksum contained in the Internet header, allowing the timestamp to be modified as the message travelled through gateways. This initial timestamp mechanism has been replaced by an unchecksummed, protocol independent timestamp option (see PSPWN No. 33). Thus, the space for timestamps is allocated only if timing experiments are actually being done.

This relatively simple change had wide-spread effects. Every program which interpreted Internet packets had to be changed. This included TENEX TCP, TCP0, TCP11, both ends of the XNET debugger, the BBN PTIP gateway, all PR net station gateways and Satellite network gateways, and the various Internet bootstraps and support programs such as the BBN line printer system. The change was made with only minor problems which were resolved within a week.

VI. PR NET RESEARCH AND DEVELOPMENT ADVANCES

A significant part of our effort bears upon the research and applied theory of the Packet Radio Network. In this section we discuss some of the problems which relate more to protocol and algorithm design than to implementation.

A. PR Net Protocol Design

An effort was made to further the development of point to point routing which had been earlier described in PRTN 194. A meeting was held between BBN and Collins Radio Group personnel to clarify the design and to discuss unresolved issues which had been raised concerning that design. The chief unresolved issue was the use of 8-bit selectors for identifying successive PRs on the route of a packet. The decision at the meeting was to implement the design in PRTN 194, however, discussions continued in the ensuing weeks due to the limitation on network size imposed by the use of an 8-bit selector. Strategies were developed such that the limitation is imposed on the number of repeaters under the control of one station but not on the number of terminals nor on the total number of repeaters in a multi-station network. At the termination of the quarter, these issues had not been fully resolved.

Also discussed at this meeting was the use of neighbor tables by the station to help plan routes to avoid congested areas. These ideas were first presented by BBN in PRTN 183. The

principle is to use counts of packets received by a PR from a neighbor and counts of packets transmitted from that neighbor to produce a measure of the fraction of packets successfully received. This, in turn, should yield a measure of the probability that a packet transmitted from the neighbor will be heard by the PR. Since that probability directly affects the delay in transmitting a packet across that hop, it is an important parameter to use when selecting routes. Agreement was reached on the principle involved, but details were not worked out.

The concept of a "local ROP" was discussed. A ROP is a packet transmitted by a PR to let everyone know it is alive. Currently, each such ROP is forwarded to the station by each PR hearing it. This results in a great volume of traffic for the station to process. The alternative is for each PR to accumulate a table of what neighboring PRs can be heard and then transmit that table to the station when significant changes in its contents occur as well as periodically. The former gives the station connectivity information and the latter gives the station information on which PRs are alive. The potential for reducing packet traffic due to ROPs is great.

B. Measurement of Station ROP-Processing Bandwidth

Every PR periodically emits ROPs (Repeater-On-Packets), which are forwarded to the station labeling (or control) process by each PR that hears them. The labeler uses the ROPs to find

out what connectivity exists. The ROP frequency must be low enough to avoid becoming the main traffic in the net, but high enough to keep the station's picture of network connectivity current so that it can keep PR labeling current. As the number of PRs at SRI increased, thus increasing the rate of ROPs arriving at the station, it became important to know what the station's ability to cope with ROPs was.

To get some feeling for the station's ROP-handling capacity, we performed a series of experiments. The station was run in a two-PR network with no traffic other than ROPs. Various ROP arrival rates were achieved by setting the ROP intervals in the PRs, choosing whether the PRs were labeled or unlabeled, and patching the PR software. The measurement process was used in the station to collect and print cumulative statistics from the connection process (telling how many packets were received and how many were dropped) and the label process (telling how many ROPs were received). The experimental method is described in more detail in PRTN 217, "Measurements of Station ROP-Processing Capacity."

The experiments and measurement data are tabulated in PRTN 217, along with a discussion of the interpretation of the results. The maximum number of ROPs the station could handle was about 41 per second. However, the actual number depends very much on what else is going on in the station.

C. The Hop Acknowledgement Protocol

This quarter we found a bug in the hop acknowledgement protocol used to improve transmission reliability in the PR net. We worked out the solution with Collins, but they have not yet implemented it. It is expected to appear in the next PR software release.

In the rest of this discussion, we refer to the PRs in the following diagram.

```
level:          L     L+1   L+2   L+3

                       B
   PR:          A            C     D
                       E
```

If B transmits a packet which is routed to C then D, it decides the transmission to C succeeded (and thus stops retransmitting) when it hears the packet transmitted by C (to D). This is a passive hop acknowledgement (hop ACK); C took no special action to provide an ACK to B. If, however, B transmits a packet which is routed only as far as C, then there is no transmission by C for B to hear, so C explicitly transmits the packet header for B's benefit. This is an active hop ACK. C also provides an active ACK if the transmission to D will be at a different data rate than the reception from B was.

How does B recognize that a packet it receives is a hop ACK for one it is sending? Basically, one packet may be a hop ACK for another if it has the same Unique Packet Identifier (a subset

of the header). However, that is not enough. After all, if the PR (A) that sent the packet to B retransmits it while B is meanwhile forwarding it to C, we don't want B to mistake a retransmission it receives from A as a hop ACK from C.

Every PR has a (hierarchy) level. Every packet has a current hierarchy level in it, which is the level of the PR that should accept it. This packet level is incremented [decremented] for each outbound [inbound] hop the packet traverses. To simplify the discussion, let's assume an outbound packet; the same holds true for inbound packets.

When a level L PR accepts a level L packet, it increments it to level L+1 before forwarding it. Since the passive hop ack for this is transmitted by the level L+1 PR, it has level L+2. Active hop ACKs, however, do not have their level incremented. (Incrementing it would have the disadvantage that PRs could not be placed at the highest (or lowest) possible level, since such PRs could not increment the level field without its overflowing.) A packet is thus accepted as a hop ACK if its level is greater than L (less than L for inbound packets).

The bug in this is that active ACKs from the next level cannot be distinguished from transmissions at the same level. For example, assume A alternate routes a packet, so that both B and E are transmitting it to C. If either hears the other's transmission, it will erroneously treat it as a hop ACK from C, since the level is L+2 (greater than L+1).

The solution to this problem is to increment the level for active hop ACKs (and not place PRs at the highest or lowest level), and require that the hop ACK be at a level greater than the PR's level+1. But this is not sufficient.

If C receives a packet from B for forwarding to D, but active ACKs it (because it is changing data rates), D cannot distinguish the active ACK (at level L+3) from the real packet (also at level L+3). It may thus accept the textless ACK instead of the real packet. There is no such problem with passive ACKs, since a passive ACK is the real packet. Thus in addition to the protocol change mentioned above, active ACKs will be explicitly marked as such (by a new header bit). Active ACK packets will never be treated as regular packets for processing or forwarding, only as hop ACKs.

D. PR Buffer Lockup

We observed a problem in the handling of packet buffers in the PR, and suggested a solution which Collins then implemented. The problem was that once a terminal (or station) PR queued a packet for transmission to its attached terminal (or station), the packet remained queued until it was transmitted. If the terminal was down, and thus not reading from its PR, the packets would remain queued indefinitely. Thus it was possible for all of a PR's buffers to become stuck on this queue, preventing the PR from accepting or transmitting any radio packets. Even if only one packet got stuck, it would prevent the PR from emitting

ROPs, since PRs only ROP if they have no packets queued for transmission. This was the case we observed -- a PR which forwarded ROPs it heard and accepted labels and debugging packets, but which did not emit ROPs.

The solution which was adopted was to time out packets on the terminal/station queue, dropping any packet that remained on the queue more than some number of seconds. SRI pointed out a further advantage of this timeout, namely to prevent packets from existing in the net indefinitely. Various network protocols depend on the concept of a maximum packet lifetime.

The process of identifying this problem was also of interest. The symptom (the non-ROPing PR) showed up in the SRI net while the SRI station was being run remotely from BBN, and the problem was found using XRAY in the station to examine the problem PR. This is an example of the usefulness of cross-radio debugging and a demonstration of the ability to solve a real PR net problem via the ARPANET and PR net from across the country.

VII. SUPPORT

The effort to develop and deliver station software is made possible by a substantial battery of support software and hardware. Without this support, development of station software would be slow or impossible. Hence we have continued to maintain the support facilities, as described below.

A. XNET Support

Several enhancements to XNET were made during this quarter to support the disk loading facility in the packet radio station. These changes revolved around the need to be able to manipulate address spaces explicitly and the need to read and write programs on the station disk. To acheive this, the field of debugger messages which previously carried a process ID was redefined to carry either a process ID or a VSM (virtual storage map) ID. To distinguish between the two, the sign bit of that field is set if a VSM ID is present. The create process primitive either creates an address space and returns the VSM ID or creates a process and returns a process ID. The latter happens when using a bootstrap program or if a VSM ID is supplied with the "create" debugger primitive. The process is created in the specified address space.

When an address space is being debugged, its contents may be written onto a specified disk segment. Later, the contents of that disk segment may be read back into an address space. Along

with the contents of the address space is written information which enables the exact state of that address space to be restored. This information consists of the bounds of the program loaded into that address space, the bounds of the pages defined for that address space and the start addresses of all processes which are to run in that address space.

XNET commands were added to facilitate certain operations which used to require a number of commands and which occur very commonly in practice. An example of such a combined action is to switch attention to another process. To do this, first the current process must be halted if it is running, then breakpoints must be removed, then the process continued if it was previously running. Then the process is end-debugged, and the new process debugged. This sequence is replaced by the "n;D" command which does all of the above and ends up debugging process n. Another example is the n,m;F command which restarts the machine at location n and sets the flags in location 46 to m. Here the equivalent sequence is to end debug the current process or address space and debug address space 0 as in the ;D command. Then the new switches are deposited into location 46 and a process created which starts running at location n. Finally, a NOP command is sent to verify that the system has successfully restarted.

B. MACN11 Support

During the course of transferring the packet radio development effort onto the DEC TOPS20 system, a number of improvements were made to MACN11 to speed the production of software. The principal change was to permit compiling and loading PDP-11 programs through the compile-class commands of the EXEC. MACN11 was converted to operate in this mode and the necessary changes were fed back to ISI to be incorporated as part of the standard MACN11 distribution.

C. Hardware Support

Half way through this quarter an I/O channel PC board failed in Packet Radio Digital Unit 2. The board was returned to Collins Radio, and they promptly sent a replacement which was installed and is working.

During this quarter we have been anticipating the delivery of several pieces of hardware to upgrade the station 2 PDP-11. The hardware on order includes:

* An IMP11-A interface to connect station 2 to the BBN RCC net (and thence the ARPANET) for loading, cross-net debugging, gateway traffic and measurement traffic.

* 32 K additional core memory to enable testing and use of the full complement of station software.

* An RK05 disk drive and controller, to permit development, testing and use of disk loading ELF and measurement file spooling.

This shipment is scheduled for delivery in the next quarter, so plans have been made for where and how to integrate it with the existing hardware. Major considerations have been to:

1) maximize the utility and convenience of the two systems;
2) permit power-down status of either system while the other system remains usable;
3) enable easy changing of disk connection from one system to the other; and
4) follow good engineering practice with regard to Unibus loading, cable length, and physical arrangement.

We have prepared a hardware layout which meets these constraints and is shown in the accompanying figure. The Unibus loads and cabling are as shown, and the disk is in effect switchable from one system to the other by simple recabling.

BBN PACKET RADIO HARDWARE LAYOUT

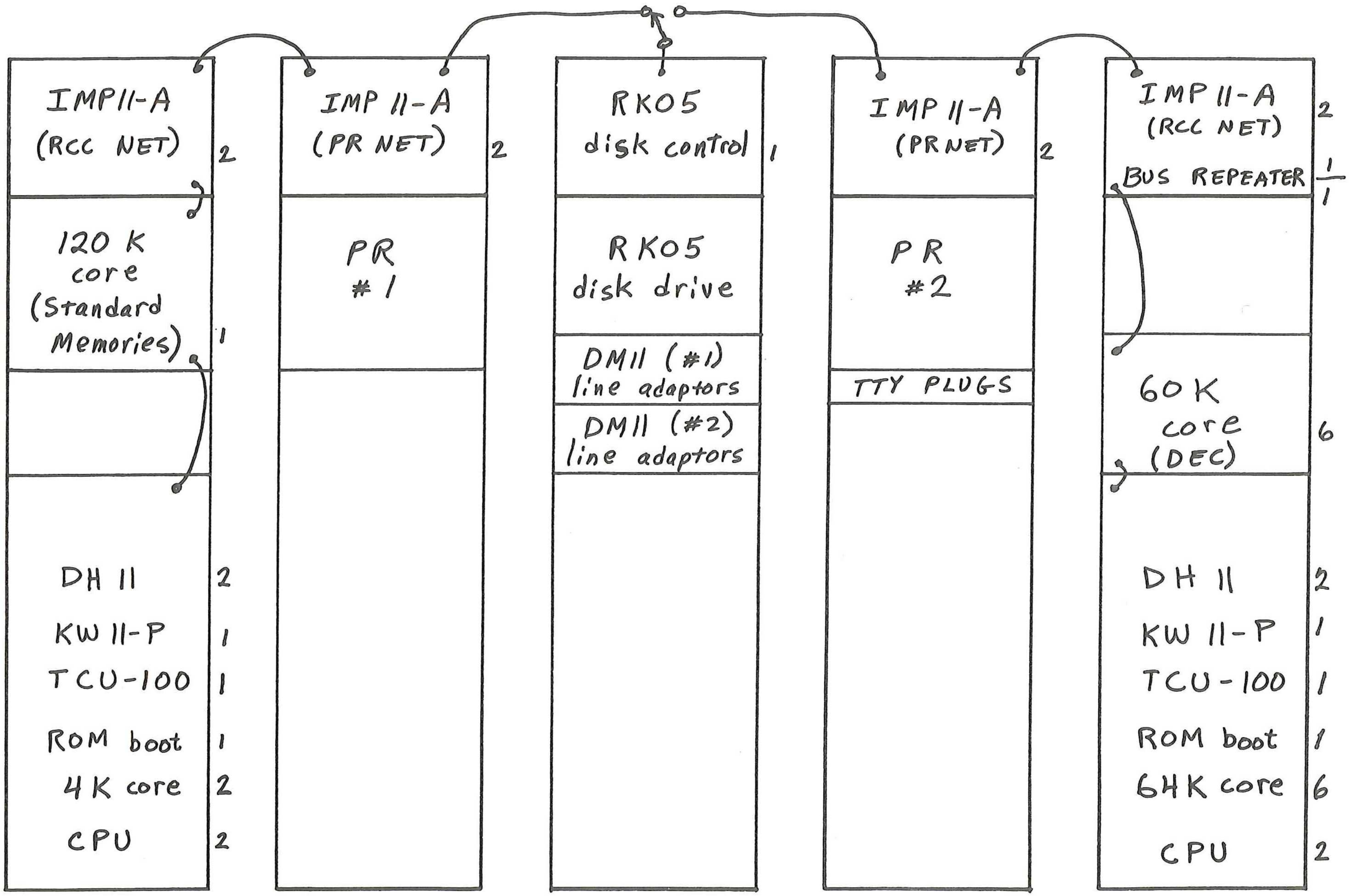


TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	PERCEPTUAL-MODEL-BASED VFR SCHEME	3
	A. Transmission Error Computation	4
	B. Parameter Quantization	8
	C. "Look-Ahead" Procedure	11
	D. "Back-Up" Procedure	11
	E. Flow Chart of the VFR Scheme	13
	F. Experimental Results	17
	G. Simplified VFR Scheme	22
III.	REAL-TIME IMPLEMENTATION	26
IV.	SPEECH QUALITY EVALUATION	27
V.	REFERENCES	50

I. INTRODUCTION

During this past quarter we completed the development of an improved automatic variable frame rate (VFR) scheme for transmitting LPC data based on our perceptual modeling approach. This involved improvements and simplifications to the distance or error measure employed in our previous version of the automatic scheme, and several refinements in the transmission strategy. We also modified the algorithm to account for the effect of parameter quantization. From an analysis data rate of 100 frames/sec (fps), the improved VFR scheme transmits an average of about 25 fps (or about 2 frames/phoneme) while producing speech quality equivalent to that of the full 100 fps system. After completion of the design of the algorithm, we experimented to determine the effect of removing the more complex parts of the scheme on speech quality and transmission frame rate, with the motivation to facilitate real-time implementation of the scheme on existing fast signal processing systems within the ARPA community. The results of these experiments showed that the tested simplifications yielded only a slight increase in transmission frame rate and produced no significant change in speech quality.

We developed the necessary software package for computing a number of our previously developed objective speech quality measures and for determining the correlation between the objective speech quality scores produced by a given objective

measure and the corresponding subjective quality results. In the next quarter, we plan to extensively test the validity of some of our objective speech quality measures, using the results of our recent subjective speech quality tests as a basis.

In our work towards a real-time LPC system, we implemented the latest version of EPOS (Environment for Processing of On-line Speech, supplied by ISI) on our PDP11. We also implemented an interactive playback program on the PDP11, which allows the user to easily specify and play out any sequence of digitized speech signals.

In our speech quality evaluation work, we analyzed the rating data from our factorial subjective quality study (described in QPR No. 7 and QPR No. 9) using the INDSCAL multidimensional scaling program. This model accounted for 74% of the variance in the data, and yielded four interpretable dimensions.

In the last quarter, we presented our results from the perceptual model work and from the analysis of our latest subjective speech quality test data, at the IEEE International Conference on Acoustics, Speech, and Signal Processing [1,2] and also at the June ARPA meeting.

II. PERCEPTUAL-MODEL-BASED VFR SCHEME

A detailed description of our perceptual model of speech and a two-stage automatic VFR scheme based on this model has been previously given [1,3]. Below, we briefly summarize only those aspects that directly relate to subsequent discussions.

The VFR scheme based on our perceptual model transmits only those frames of LPC data, out of the 100 fps analysis data, which are necessary to render the resulting speech quality essentially indistinguishable from that of the unreduced 100 fps system. For example, it transmits more frequently during rapid speech transitions and less frequently during steady-state portions of speech. The scheme explicitly uses the hypothesis of the model that LPC parameters vary linearly between adjacent transmitted frames. Transmission decisions are made independently for pitch, gain and log area ratios (LARs). For pitch and gain, we presented in the last Quarterly Progress Report an improved VFR scheme called FIT or Fan Interpolation Technique [3]. A two-stage automatic VFR scheme was previously developed for transmitting LARs [1,3].

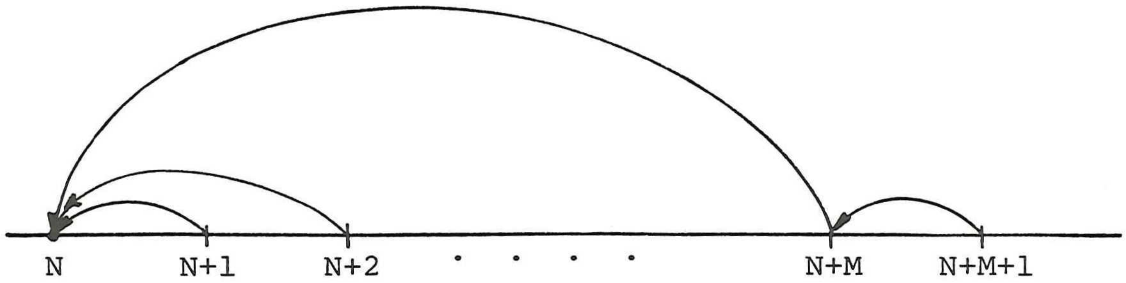
A major difference between the perceptual-model-based VFR

scheme and our earlier VFR scheme that is being used in ARPA LPC-II system [4] is in the transmission strategy: our earlier scheme performs an "end-to-end comparison," illustrated in Fig. 1a, between the preceding transmitted frame and the current frame being considered for transmission; on the other hand, the new scheme as shown in Fig. 1b, compares LPC parameters of <u>every</u> frame in the transmission interval with those obtained by linear interpolation between the two "end-frames" and computes the total transmission error as some weighted average of the individual frame errors. It is this difference which has led to a substantially lower transmission frame rate for the new scheme than for our earlier scheme.

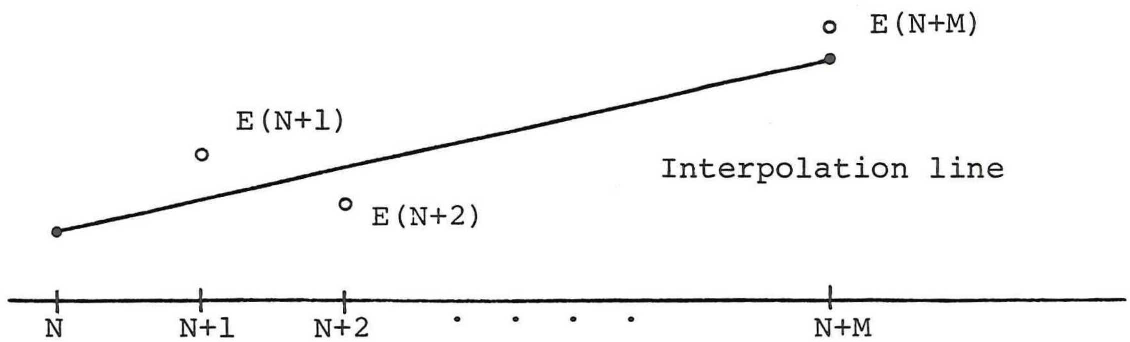
Below, we report on several modifications that we made during the last quarter on the two-stage VFR scheme for the transmission of LARs.

A. Transmission Error Computation

Given that LARs of the frame N, say, have been transmitted, the basic strategy is to determine the longest line extending from $\underline{g}(N)$ (vector of p LARs for frame N) in the p-dimensional parameter space such that the resulting transmission error computed between the actual parameter vectors $\underline{g}(N+i)$ and the interpolated parameter vectors $\underline{\hat{g}}(N+i)$ over the duration of that line is less than some threshold (see Fig. 1b). First, we need to define frame error, or distance between two sets of LARs $\underline{g}$ and $\underline{\hat{g}}$ for any given frame, and then specify how this error is averaged over several frames (time averaging).



(a)



(b)

Fig. 1 Illustration of two VFR transmission strategies.

(a) "End-to-end" error measurement of our old VFR scheme used in LPC-II.

(b) Average frame error between actual and interpolated values computed over the transmission interval. E(N+i) is the error for frame N+i. The frame error E(N+M) is due to parameter quantization (see Section II-B).

Frame Error:

Frame error for frame n denoted by E(n), is defined as the weighted Euclidean distance:

$$E(n) = \sum_{i=1}^{m} w_i [g_i(n) - \hat{g}_i(n)]^2 / \sum_{i=1}^{m} w_i \quad (1)$$

where $\{w_i\}$ is the set of coefficient weights chosen to reflect the relative importance of the different LARs (presumably to perceived speech quality); we allow $m \leq p$.

We have chosen the coefficient weights to be the expected or average spectral sensitivities of individual LARs. We had previously computed these average spectral sensitivities [5]; for the first 4 LARs, these are: 1.3, 1.2, 1.1 and 1.0. This weighting scheme is based on the reasonable idea that a given amount of error in a LAR with a higher sensitivity is more important to spectral accuracy (and hence perception) than the same error in a LAR with a lower sensitivity. Surprisingly, however, our experimental results showed that different choices of these weights (e.g., w_i all equal to 1) produced no discernible change in speech quality.

We found through experimentation that the summation in the frame error definition (1) need be done only up to the first 4 LARs (i.e., m=4).

Another way to compute the frame error is via log likelihood ratio measure, which is being used in LPC-II [4]. Our experiments (see Subsection F) indicated identical speech quality results for the same average transmission frame rate, for the two measures: LAR distance and log likelihood ratio. Since LARs are being used as transmission parameters, use of the LAR distance measure is computationally much less expensive than the log likelihood ratio measure. So, we have employed the LAR distance in all our subsequent experiments.

Transmission Error:

Transmission error ET between frames N and N+M is computed as the weighted, time-averaged frame error:

$$ET = \frac{1}{M} \sum_{n=N+1}^{N+M} W(n)\ E(n)\ , \tag{2}$$

where W(n) is the frame weight for frame n. (The upper limit for the summation in (2) is considered as N+M to incorporate the effect of LAR quantization; E(N+M) is computed from (1) with $\hat{g}_i$ denoting quantized LAR values.) As frame weight, we have successfully used the speech signal energy per sample in that frame, R0, expressed in decibels and normalized with respect to some estimate RM of the maximum value of R0:

$$W(n) = R\emptyset(n)/RM(n). \tag{3}$$

The idea behind the weighting scheme given by (3) is that even large frame errors do not make perceptible effect if they are associated with relatively small speech signal energies. For our speech data base, where we have 9-bit samples, R0 is usually around 35-40 dB for open vowels, 15-30 dB for fricatives, and around 0-7 dB for the silent period of an unvoiced plosive.

A simple and efficient way to update RM is by the following recursive method:

$$RM(n) = \text{Max}\,\{R\emptyset(n),\ \alpha\, RM(n-1),\ 25\text{ dB}\}, \qquad (4)$$

where α is a constant less than 1. We use $\alpha = 0.98$, which means that RM decays to half its original value in about 27 frames. It should be noted from (3) and (4) that $W(n)=1$ if $R0(n)>25$ and has been increasing or has been decreasing slowly; $W(n)<1$ if $R0(n)<25$ or if $R0(t)$ has been decreasing at a faster rate than $\exp(0.98t)$.

B. Parameter Quantization

There are two ways in which the effect of parameter quantization can be included within the above procedure for transmission error computation. Both ways can be employed simultaneously.

First, since the transmitted LARs have to be quantized, we consider the interpolation line between the quantized LARs of the two end-frames (frames N and N+M in (2)). A frame error is then computed as distance given by (1) between the unquantized LARs of that frame and the corresponding LARs obtained from the above interpolation line. The frame error for the right end-frame (E(N+M) in (2)) is entirely due to parameter quantization.

The second way of incorporating parameter quantization is what we call the "adjustable" quantization method. A parameter value is normally quantized to its nearest quantization level. The adjustable quantization scheme allows either of the two nearest quantization levels. Thus, given the quantized LARs of the initial frame (left end-frame), the scheme determines the adjusted quantized values of the LARs for the final frame (right end-frame) in the transmission interval, in such a way that the total transmission error is minimized.

A one-dimensional (p=m=1) example is shown in Fig. 2 to illustrate the "adjustable" quantization. For this example, the parameter value of the sixth frame is selected for transmission. If this value is quantized to the nearest quantizer output (the output just below it), there is considerable interpolation error in the interval between frames 1 and 6. If the higher quantizer output is used instead, the total transmission error is reduced. (Fig. 2 also shows the interpolation line for the next transmission interval from frame 6 to frame 11.)

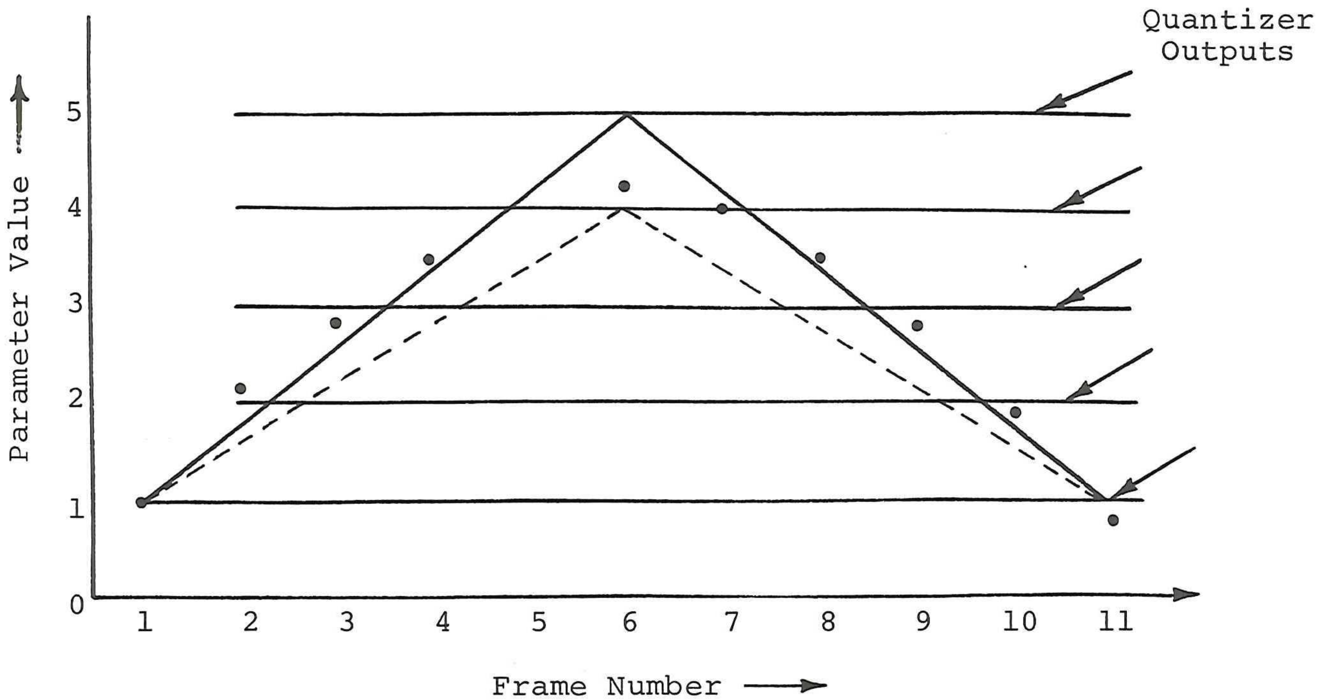


Fig. 2 Example to illustrate the "adjustable" quantization scheme. Dashed-line plot corresponds to normal quantization, where a parameter value is quantized to the nearest quantizer output. Solid line corresponds to the "adjustable" quantization (see text). (The dots represent the original unquantized parameter data.)

C. "Look-Ahead" Procedure

Sometimes the transmission error may temporarily exceed the prespecified threshold. However, if the transmission interval is lengthened, the error may drop below the threshold. An example is illustrated in Fig. 3. In Fig. 3a, the first and third frame values are considered to be transmitted; in Fig. 3b, the first and the fifth frame values are shown as being transmitted. The transmission error for the case in Fig. 3b is seen to be lower than for the case in Fig. 3a.

We call the above feature a "look-ahead" feature. The extent of "look-ahead" (in terms of number of frames to consider) is limited only by the resulting computational burden; we use a four-frame "look-ahead" procedure. If the error does not drop below the threshold even after moving forward by four frames, we hypothesize the transmission of the frame immediately preceding the one where the threshold was first exceeded.

D. "Back-Up" Procedure

Once we have determined three successive transmission frames which will keep the transmission errors in the two transmission intervals below the threshold, we then reposition the middle transmission frame by backing up, in order to minimize the total

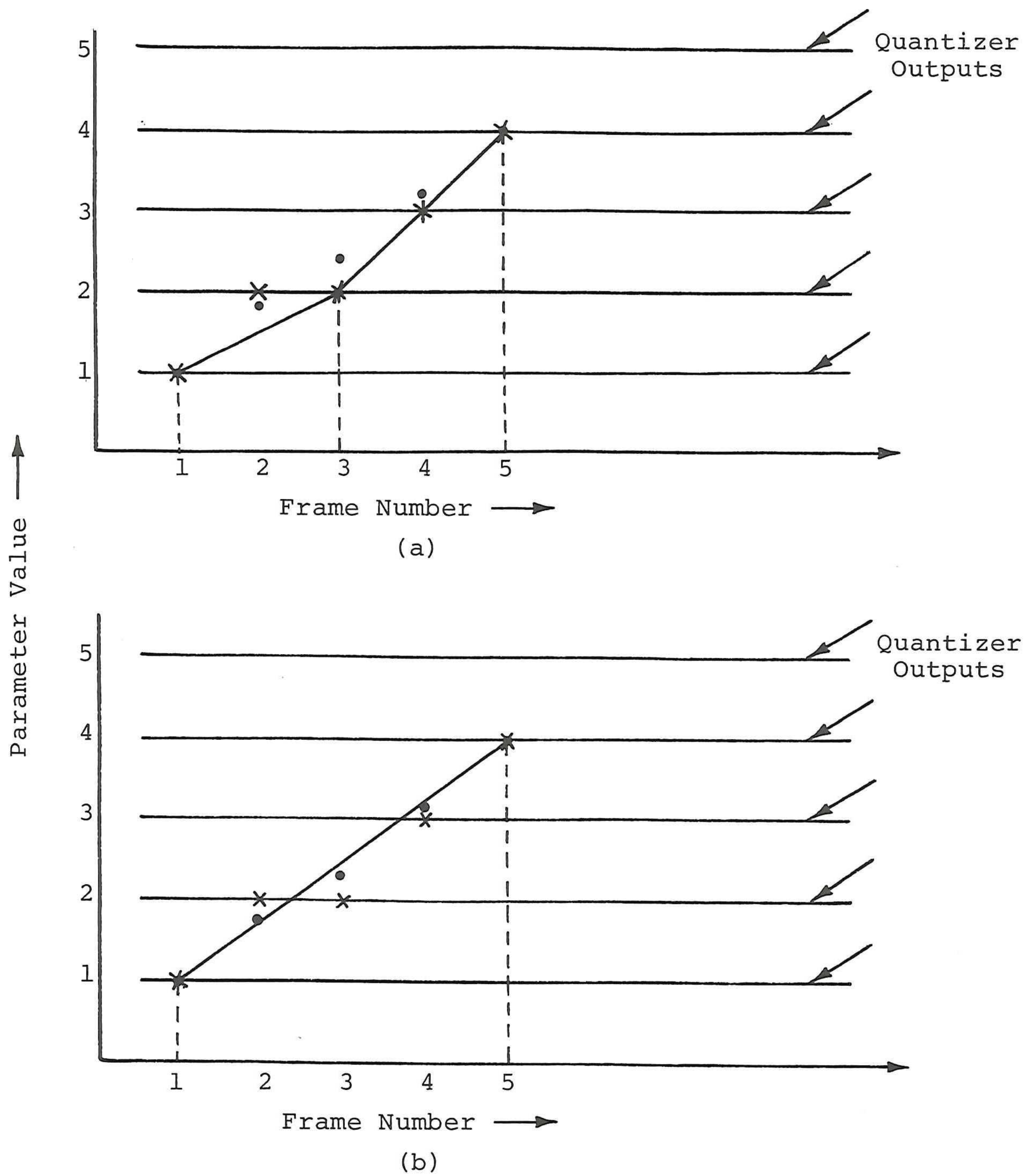


Fig. 3 Example to illustrate "look-ahead" procedure. The dots represent the original unquantized parameter values. The x's represent quantizer output values. The vertical dashed lines indicate frames chosen for transmission.
(a) Without the "look-ahead" scheme
(b) With the "look-ahead" scheme

transmission error in both intervals. (When using the above "adjustable" quantization with the "back-up" procedure, we compute the total error in the two transmission intervals by first computing the "adjusted" quantized values for the second and third transmission frames. This is illustrated in Fig. 2, where the three successive transmission frames considered are frames 1, 6 and 11.)

Fig. 4 illustrates the "back-up" procedure by way of an example. The VFR scheme initially decided to transmit frames 3, 8 and 13, as shown in Fig. 4a. The two interpolation lines are also shown. Fig. 4b clearly demonstrates that if frame 7 were transmitted instead of frame 8, the interpolated values would match the original data much more closely in both transmission intervals.

E. Flow Chart of the VFR Scheme

The flow chart of the VFR scheme described in the previous sections is given in Fig. 5. Variables that appear in the flow chart are defined in Table 1. A function called ERROR is used to compute the transmission error between two hypothesized transmission frames. It accepts as input, quantization levels for the LARs at the first or initial frame, and determines the "adjusted" set of quantization levels for the second transmission frame. If the function is called with three transmission frames, it provides the optimal set of quantization levels for the second and third transmission frames. Each box shown in the flow chart translates into one or two FORTRAN statements.

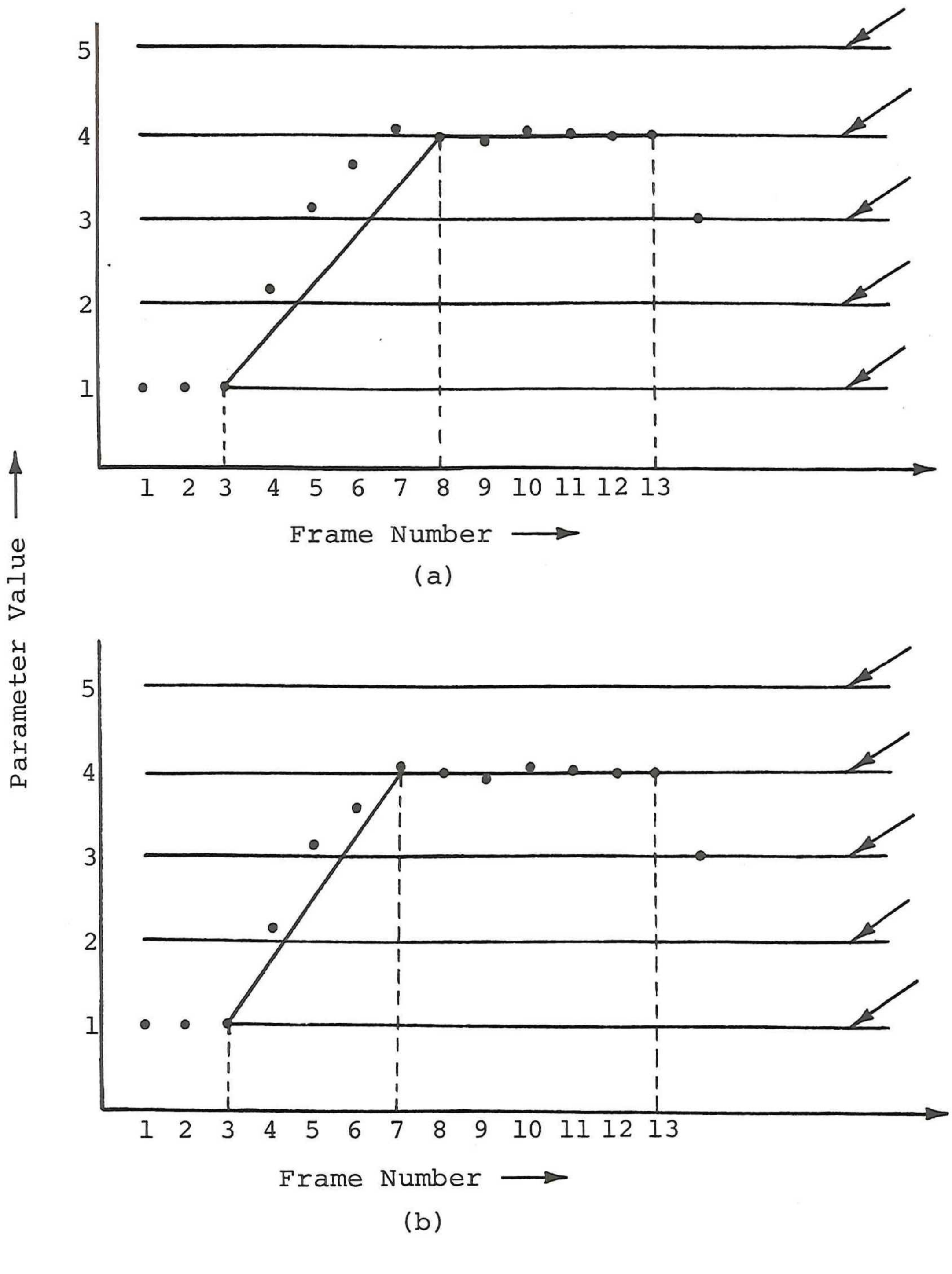


Fig. 4 Example to illustrate the "back-up" procedure. The dots represent the original unquantized parameter values. The vertical dashed lines indicate frames chosen for transmission.
(a) Without the "back-up" scheme
(b) With the "back-up" scheme

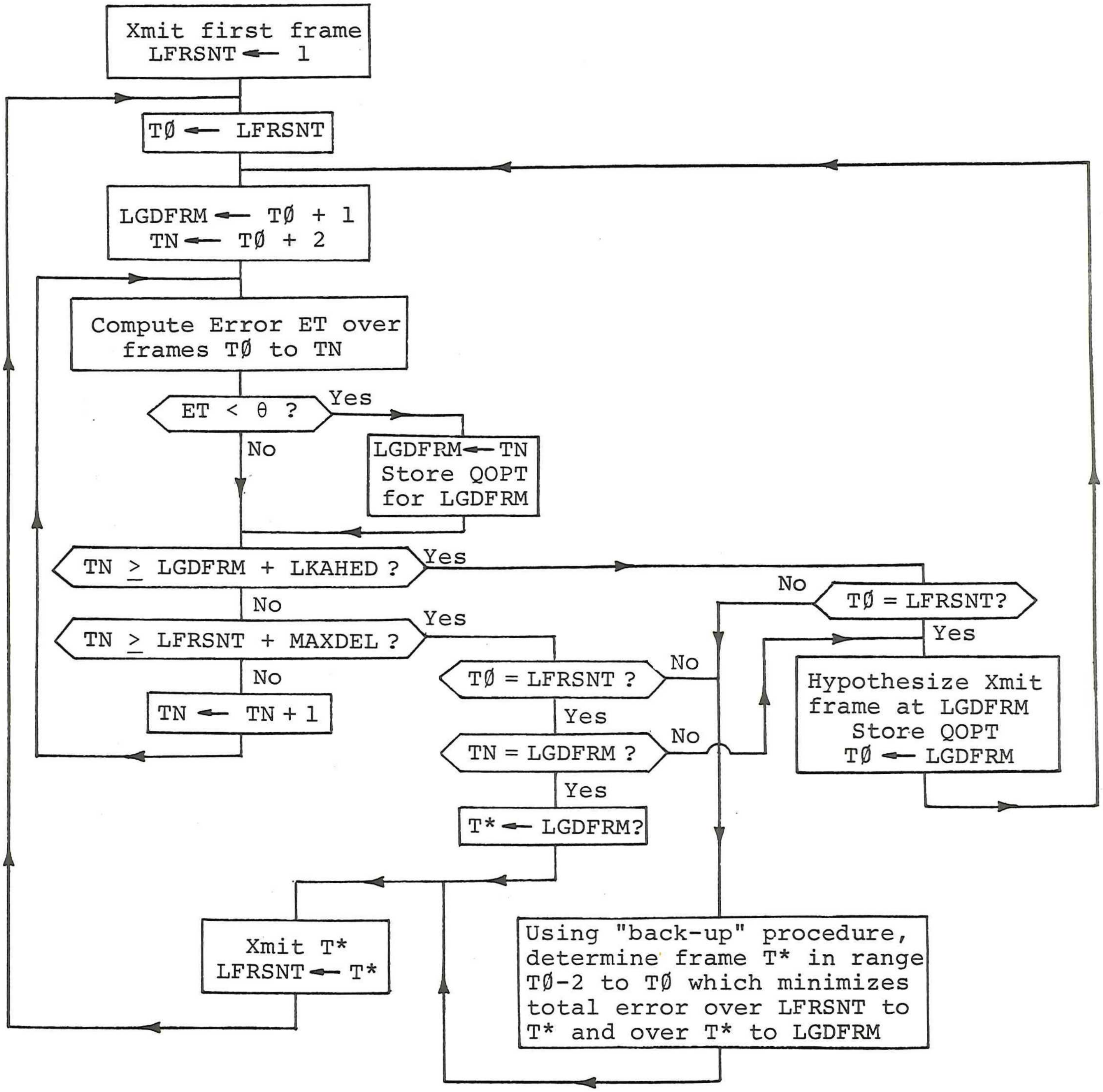


Fig. 5 Flow chart of full algorithm.

Table 1

List of Variables Used in the Flow Chart in Fig. 5.

LFRSNT	Last frame actually transmitted.
TØ	First frame (left end-frame of the interpolation line) in a transmission interval. TØ equals either LFRSNT or a hypothesized transmission frame when using the "back-up" scheme.
TN	Current frame (right end-frame of the interpolation line).
ET	Transmission error between the original unquantized LAR data, and the quantized and interpolated values, computed over the interval from frame TØ + 1 to frame TN (see eq. 2 in the text).
θ	Transmission error threshold. Normally, θ=1.3
LGDFRM	Last good frame, i.e., last frame where ET < θ.
LKAHED	Number of frames to "look ahead" beyond the frame where ET exceeds θ. Normally, LKAHED = 4 frames.
MAXDEL	Maximum allowed transmission delay. Without the "back-up" scheme, it is the maximum transmission interval permitted. With the "back-up" scheme, it is the maximum allowed interval between a transmitted frame (LFRSNT) and a frame which is the second hypothesized transmission frame after LFRSNT if it is not LGDFRM, or the second hypothesized transmission frame plus LKAHED, otherwise. Normally, MAXDEL = 12 frames (12Ø ms).
QOPT	Quantized LAR values resulting from the "adjustable" quantization scheme.
T*	Frame position, determined by the "back-up" procedure.

F. Experimental Results

We tested the VFR algorithm described above on a set of nine sentences (JB1, DD2, RS3, AR4, DK4, JB5, RS6, DK6 and DD6; 6 sentences from 3 males and 3 sentences from 2 females) from the data base used in our speech quality evaluation work. Table 2 describes six vocoder systems and lists their average transmission frame rates and bit rates obtained over the nine sentences. We ran informal, pair-wise speech quality comparison tests on the syntheses from these six vocoder systems, to evaluate the relative performance of the different versions of the above VFR scheme.

Vocoders 1 and 2 given in Table 2 employ the full 100 fps fixed-rate transmission for all parameters (pitch,gain and LARs). Vocoder 1 uses the unquantized parameters for synthesis, while Vocoder 2 quantizes the parameters using 5 bits for gain, 6 bits for pitch (plus 1 bit for Voiced/Unvoiced status), and 44 bits for LARs of voiced frames and 42 bits for LARs of unvoiced frames, which results in a transmission bit rate of about 5650 bps. Vocoders 3-6 quantize the parameters in the same way, but employ VFR transmission for all parameters. For pitch and gain VFR transmission, they all use the double-threshold FIT scheme [3] on the quantized values (levels), with thresholds of 0 and 1 for pitch, and 1 and 2 for gain; this yielded a transmission frame rate of about 28 fps for pitch and 24 fps for gain. The VFR scheme used for LAR transmission becomes progressively complex going from Vocoder 3 to Vocoder 6, with Vocoder 6

Vocoder System	Parameter Quantization	Adjustable Quantization	Look Ahead	Backup	Pitch & Gain Fixed or Variable Frame Rate	LAR Frame Rate (fps)	Average Bit Rate (bps)
1	No	-	-	-	Fixed	100 (Fixed)	-
2	Yes	No	No	No	Fixed	100 (Fixed)	5650
3	Yes	No	No	No	Variable	30	1850
4	Yes	No	Yes	No	Variable	27	1750
5	Yes	No	Yes	Yes	Variable	27	1750
6	Yes	Yes	Yes	Yes	Variable	25	1650

Table 2. Description of six vocoder systems tested and their average transmission frame and bit rates.

employing the complete VFR scheme described in the last section via flow chart. The simplest VFR scheme (used by Vocoder 3), employs the quantized LARs of the end-frames of the interpolation line (see Section II-B). For all the four vocoders, the threshold θ (see flow chart in Fig. 5) for the transmission error ET was chosen as 1.3. (We chose m=4 in the expression (1) for frame error since it yielded the same speech quality as any higher value but at the least computational effort.) The above choice of the transmission error threshold produced an average frame rate of about 25 fps for the full scheme (Vocoder 6) and an average transmission error (ET averaged over the nine sentences) of 0.55.

Informal tests of pair-wise speech quality comparisons were run for the six vocoders. Also, we compared the full VFR scheme (Vocoder 6) with our earlier "end-to-end" scheme used in LPC-II and with the 50 fps fixed-rate scheme used in LPC-I. (The latter two vocoders we considered were <u>not</u> LPC-II and LPC-I in view of the differences in vocoder conditions such as speech signal sampling rate, bit allocation for parameter quantization, and pitch extraction scheme.) Below, we describe the results of only the important comparisons. (Speech quality tests comparing Vocoders 3-5 with Vocoder 6 are given in Subsection G.)

1. Vocoder 2 vs Vocoder 6. There were cases for which speech transitions were more "crisp" for Vocoder 2 (5650 bps) than for Vocoder 6 (1650 bps). However, for most sentences

(especially the slowly varying ones, JB1 and DD2), the synthesized speech from Vocoder 2 sounded worse in that it had appreciably more "wobble" quality than the synthesis from Vocoder 6. Our explanation for the observed quality difference is that for the cases when the "wobble" quality is perceived, the error due to parameter quantization is more than the error due to parameter interpolation.

2. Same comparison as in (1), except that both systems used unquantized parameters in the synthesis (i.e., Vocoder 1 vs unquantized version of Vocoder 6). The syntheses for the slowly varying sentences JB1 and DD2 from the variable rate system had slightly less "wobble" quality than those from the fixed rate system. This is probably due to the fact that small inaccuracies in the LPC analysis arising from interaction between the pitch period and the analysis interval tend to be reduced by the smoothing effect of the interpolation employed by the VFR scheme. There were a couple of situations (during the part "trouble with" in the sentence DK6) where the fixed rate synthesis sounded better. In general, Vocoder 1 and the unquantized version of Vocoder 6 produced speech with essentially the same quality.

3. Vocoder 1 vs Vocoder 6. Surprisingly, the results of this comparison between the unquantized 100 bps system and the 1650 bps VFR system were the same as given above in (2).

4. Vocoder 6 (1650 bps) produced speech quality equal to or better than that of the VFR system with the earlier "end-to-end" scheme of LPC-II (2100 bps). Speech quality improvements observed in the syntheses from Vocoder 6 included clarity and "crispness" of several syllables which were slurred when processed through the earlier VFR system.

5. Vocoder 6 (1650 bps) was compared against the 50 fps fixed-rate system (2825 bps). LPC-I also uses the 50 fps fixed-rate transmission but operates at even a higher bit rate of about 3600 bps. Although the 50 fps system had less "wobble" quality than the 100 fps system (Vocoder 2), it still had a more "wobble" quality than Vocoder 6, especially for the sentences JB1 and DD2.

6. Finally, we employed the log likelihood error measure [4] for computing the frame error between the two sets of LARs $\underline{g}$ and $\underline{g}$, instead of the weighted Euclidean distance measure given by (1). (Notice that for likelihood ratio computation, LARs are to be first transformed to predictor coefficients.) We adjusted the transmission error threshold (θ) so as to obtain about the same average frame rate (25 fps) as Vocoder 6. We found that the speech quality of the resulting vocoder was identical to that of Vocoder 6. This result leads to the following two observations. First, we conclude that the superior performance of the new perceptual-model-based VFR scheme over the earlier,

"end-to-end" scheme of LPC-II (see (4) above), is not due to the change in the definition of the frame error, but due to the difference in the way the transmission error is computed in each case (see Fig. 1 which illustrates this difference). Secondly, we recommend the use of the LAR distance measure (1) in preference to the log likelihood ratio measure, since the use of the latter measure requires about 50 times more computational time.

G. Simplified VFR Scheme

Though the algorithm described above produced very low frame rates and good quality speech, it has the disadvantage of being fairly complex, and somewhat slower than real time in our simulation on a KL-10 computer. Of course it could be coded to run in real time on a fast mini-computer, but might not leave enough time for other processing needs. Therefore, we tried several simplifications of the algorithm, in order to arrive at a reasonable compromise between speed, complexity, frame rate (and bit rate) and speech quality.

Our first simplification (see Table 2, Vocoder 5) involved the adjustable quantization. Instead of allowing two possible quantization levels for each LAR of every hypothesized transmission frame, the LAR values were always quantized to the nearest levels. This sped up the algorithm by a factor of 4, and reduced the complexity. The transmission frame rate (for the same transmission error threshold) rose to about 27 fps. However

the resulting sentences were indistinguishable from those produced by the scheme with adjustable quantization.

For the second simplification we eliminated the "back-up" procedure (Vocoder 4). The frame rate remained unchanged at 27 fps, but the average measured transmission error increased by about 20%. Careful, repeated listening through headphones revealed only a slight degradation for two sentences. The differences were not perceived through high quality loudspeakers, and were not noticed on single paired-comparisons through headphones. This simplification sped up the algorithm by another factor of 3, and reduced the complexity considerably.

The third simplification was the removal of the "look-ahead" procedure (Vocoder 3). That is, as soon as the transmission error computed over the interval from the preceding transmitted frame to the current frame exceeded the threshold, the frame immediately preceding the current one was chosen to be transmitted. As expected, this increased the frame rate substantially (to 30 fps), for the same speech quality. When the "look-ahead" procedure enabled the algorithm to skip over a bad region, the transmission intervals were greatly lengthened. The simplification reduced processing time by about 30%, and eliminated only 3 lines of FORTRAN code.

Recommended Scheme:

While the full scheme (Vocoder 6) clearly results in a lower frame rate and slightly better speech quality, it is much more complex and an order of magnitude slower than the simplest scheme (without "adjustable" quantization, and "back-up" and "look-ahead" features). The first two simplifications discussed above seem reasonable, since the resulting loss was small. The last feature ("look-ahead") is recommended, since its removal resulted in substantial losses and produced only minor gains.

Fig. 6 shows a flow chart of the recommended VFR scheme (Vocoder 4). Comparison of this simplified scheme with Fig. 5 will make the difference in complexity apparent.

Of course, if the computer running the VFR algorithm is fast enough, and easy to program, it may be worth the extra trouble to implement the full scheme, which includes the features of "adjustable" quantization and "back-up".

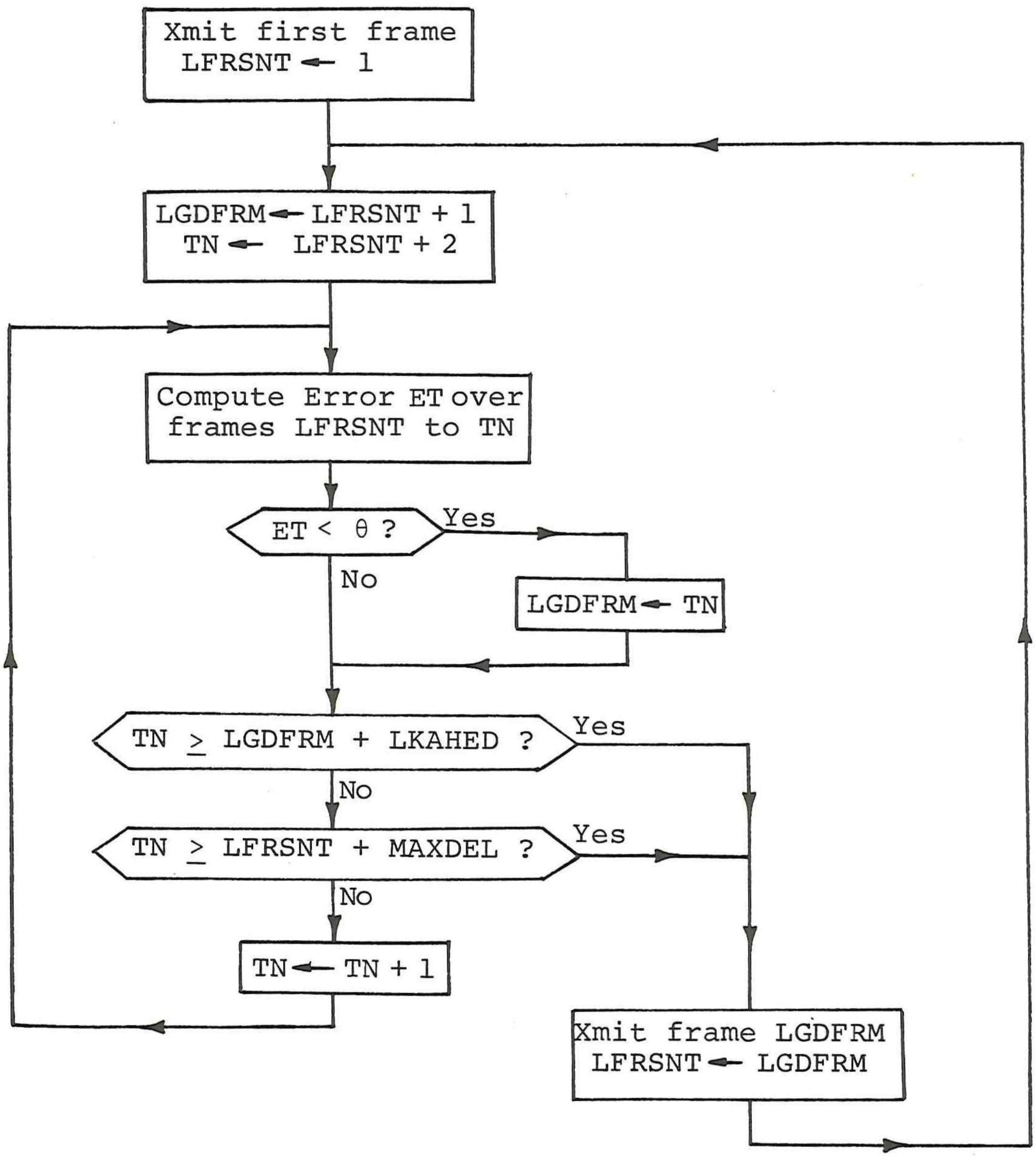


Fig. 6 Flow chart of recommended, simplified algorithm.

III. REAL-TIME IMPLEMENTATION

During the past quarter, we implemented the latest version of EPOS on our PDP11. During an attempt to run a Network Speech Compression experiment with ISI, we established a connection to ISI but were unable to transmit or receive speech due to some problems, either hardware or software, with our SPS41. We plan to continue the debugging process on the SPS41.

We implemented an interactive playback program on the PDP11, which allows the user to easily specify and play out any sequence of digitized speech signals. The program is quite useful for running informal listening tests, and for conveniently and rapidly preparing audio tapes for demo purposes and for formal subjective speech quality tests.

We also began to incorporate display facilities into our interactive program on our PDP11/IMLAC PDS-1 computer system, to allow digitization, display, editing and playback of speech waveforms. This effort will continue into the current quarter. Also, we plan to modify our File Transfer Protocol (FTP) program to permit multiple transfer of files from TENEX to the PDP11.

IV. SPEECH QUALITY EVALUATION

In the last quarter, we have applied multidimensional scaling analyses (INDSCAL) to the rating data collected in our recent factorial study of quality, which was described in the last two QPR's. This has led to several results:

1. The four-dimensional INDSCAL solution accounted for 74% of the total variance in the judgments, with correlations ranging from 0.60 to 0.95 between observed data and the model, for single passes through the stimuli by single subjects.

2. The four dimensions of the solution were identified as the effects of

   a) the number of poles used by the vocoder,
   b) the sex of the speaker of the test sentence,
   c) the frame rate of the vocoder,
   d) the quantization step size of the vocoder.

The INDSCAL Model

We present a brief description of the INDSCAL model here; more detail can be found in [6,8]. The input to INDSCAL consists of two or more matrices of proximities, all pertaining to the same set of stimulus objects (the 50 vocoder systems, in the present instance). The entries in the proximity matrices usually consist of empirical measures of the similarity of pairs of

stimuli. These are obtained, for example, by presenting all pairs of stimuli to subjects, and asking them to rate the similarity or dissimilarity of each pair. Confusion data, corresponding to the extent to which stimuli are confused in an identification task, could also be used as proximity data. Thus the entry D_{ij} in cell (i,j) of an input matrix, represents the proximity of stimulus i to stimulus j, under the conditions in which the data were collected. There may be several such matrices, one for each subject in the paradigmatic use of INDSCAL. Alternatively, the different matrices may contain data obtained from different sources, or collected under different conditions, such as the seven different test sentences in the present context.

INDSCAL was developed to model explicitly the fact that subjects may disagree dramatically in how they perceive the stimuli, and differences between the stimuli. The model assumes that all subjects use the same set of underlying perceptual dimensions, but that there may be large differences between subjects in the relative salience of these dimensions. Therefore, INDSCAL models each stimulus as a point in a "group space" of one or more dimensions. The group space represents the common perceptual dimensions that all subjects make use of. The data for a particular subject are modelled by stretching or shrinking the dimensions of the group space linearly, until they reflect the relative salience of the dimensions for that subject. Thus the INDSCAL solution consists of sets of coordinates for the

stimuli in the group space, and sets of weights that are used to deform the group space to produce the idiosyncratic spaces that model the individual subjects' data. The proximities data are modelled by the (weighted) Euclidean distances between the stimulus points in the model. That is, stimuli that are judged very similar are represented by points that are very close together in the group space. (A dimension-weight is actually applied to the square of the difference of coordinate value on that dimension, so that the dimension itself can be regarded as being stretched or shrunk by the square-root of the weight. An advantage of this formulation is that, after the solution has been normalized, the square of a weight is a measure of the proportion of the variance in an individual subject's data that is accounted for by that dimension. See [8] for more detail.) The program alternately calculates the weights, and then the stimulus coordinates, using a least-squares procedure iteratively until a specified goodness-of-fit criterion is reached.

<u>INDSCAL Analyses</u>

The data from the rating study consisted of a total of 27 judgments on each of 7 sentences, as processed by each of 50 vocoders. The data collection procedures were described in QPR No. 7, and initial analyses in QPR No. 9. Before the INDSCAL analysis could be performed, it was necessary to convert the ratings into dissimilarity matrices. We followed the procedure described in [8], and generated 34 separate dissimilarities matrices, one for each of the seven sentences, and one for each

of the 27 subject-passes through the data. For the sentence matrices, the dissimilarity between two systems was calculated as the root-mean-square, over the 27 subject-passes, of the differences in the ratings assigned to the two systems on the sentence in question. For the subject-pass matrices, the dissimilarity between two systems was calculated as the root-mean-square, over the 7 sentences, of the differences in the ratings assigned to the two systems on the pass in question.

INDSCAL outputs its solution in two forms: 1) as tables of the stimulus coordinates and weights, together with various ancillary data such as correlations between the modelled and observed data, and 2) as plots of the stimuli and weights.

Tables 3 and 4 give the weights and coordinates, respectively, for the four-dimensional solution. We in fact produced solutions in up to 10 dimensions, but the additional variance accounted for by each extra dimension above four was small, and the extra dimensions were not readily interpretable, as the first four were. Therefore, we will not discuss the solutions in more than four dimensions. The four-dimensional solution accounted for 74% of the variance in the input data. This figure is encouragingly high, considering that the input data were derived dissimilarities, calculated from rating data, rather than true dissimilarities. Collecting true dissimilarities with 350 stimuli -- 50 systems x 7 sentences -- would have been prohibitively time consuming.

FACTORIAL 444 VOCODER RAW PROXIMITIES, 7 SENT + 27 PASSES, 50 VOCODERS
NORMALIZED A MATRICES, 4-D SOLUTION OF RX346 DATA
MATRIX 1: SENTENCE AND SUBJECT-PASS WEIGHTS

	DIM 1	DIM 2	DIM 3	DIM 4
SENTENCE 1	0.3800	0.1302	-0.0999	0.5960
SENTENCE 2	0.2613	0.1689	0.2443	0.4349
SENTENCE 3	-0.0801	0.3633	0.6656	-0.0444
SENTENCE 4	0.0473	0.5191	0.6035	-0.0895
SENTENCE 5	0.7577	0.0454	0.2324	0.1056
SENTENCE 6	0.8981	0.0008	-0.0579	0.1171
SENTENCE 7	-0.0643	0.3146	0.5873	0.1255
PASS 1, SUBJ 1	0.2069	0.2545	0.1080	0.5415
PASS 1, SUBJ 2	0.1243	-0.1088	0.2376	0.5872
PASS 1, SUBJ 3	0.6577	0.0530	0.0388	0.2798
PASS 1, SUBJ 4	0.2350	0.4088	0.1306	0.2321
PASS 1, SUBJ 5	0.7985	0.1598	0.1275	-0.0508
PASS 1, SUBJ 6	0.0391	0.7126	-0.0169	0.0812
PASS 1, SUBJ 7	0.2542	-0.0017	0.5243	0.2411
PASS 1, SUBJ 8	0.2796	0.4373	0.1888	0.2771
PASS 1, SUBJ 9	0.6498	-0.2003	0.4530	0.1381
PASS 2, SUBJ 1	0.2153	0.2391	0.1879	0.5413
PASS 2, SUBJ 2	0.0040	0.0003	0.3899	0.3665
PASS 2, SUBJ 3	0.6601	0.0664	0.0361	0.2788
PASS 2, SUBJ 4	0.1931	0.6879	0.0802	0.0998
PASS 2, SUBJ 5	0.7446	0.1876	0.0364	0.0348
PASS 2, SUBJ 6	0.0792	0.7039	0.0067	0.1354
PASS 2, SUBJ 7	0.2524	0.1408	0.4102	0.1503
PASS 2, SUBJ 8	0.2703	0.4742	0.1173	0.3099
PASS 2, SUBJ 9	0.5543	-0.2210	0.5940	0.1063
PASS 3, SUBJ 1	0.2114	0.6029	0.0302	0.3142
PASS 3, SUBJ 5	0.8300	0.0473	0.1094	-0.0123
PASS 3, SUBJ 6	0.3089	0.5650	0.2015	-0.0183
PASS 4, SUBJ 1	0.3410	0.5367	-0.0478	0.3249
PASS 4, SUBJ 5	0.9500	0.0679	0.0030	-0.1233
PASS 4, SUBJ 6	0.4105	0.6535	0.1169	-0.1763
PASS 5, SUBJ 1	0.2720	0.7210	-0.0648	0.1541
PASS 5, SUBJ 5	0.8300	-0.0042	0.0496	0.0461
PASS 5, SUBJ 6	0.2179	0.7504	0.0672	-0.1238

Table 3

FACTORIAL 444 VOCODER RAW PROXIMITIES, 7 SENT + 27 PASSES, 50 VOCODERS
NORMALIZED A MATRICES, 4-D SOLUTION OF RX346 DATA
MATRIX 2: VOCODER SYSTEM COORDINATES

PQR	Ps	Q-dB	F/s	DIM 1	DIM 2	DIM 3	DIM 4
000	--	----	---	-0.2980	-0.9199	-0.4926	-0.4923
111	13	0.25	100	-0.2048	-0.1334	-0.2240	-0.2156
121	13	0.5	100	-0.1507	-0.0815	-0.1940	-0.1852
122	13	0.5	67	-0.1842	-0.0659	-0.0523	-0.1881
123	13	0.5	50	-0.1988	-0.0848	0.0064	-0.1495
124	13	0.5	33	-0.0510	-0.0201	0.1422	-0.1557
131	13	1.0	100	-0.1897	-0.0401	-0.0879	-0.0392
132	13	1.0	67	-0.1500	-0.0560	-0.0328	-0.1438
133	13	1.0	50	-0.1770	-0.0413	-0.0046	-0.1454
134	13	1.0	33	-0.0642	0.0343	0.1750	-0.1034
141	13	2.0	100	-0.0162	0.0477	-0.0159	0.1753
142	13	2.0	67	-0.0271	0.0478	-0.0044	0.1966
143	13	2.0	50	-0.0684	0.0357	0.0371	0.0543
144	13	2.0	33	-0.0620	0.0373	0.1664	-0.0083
221	11	0.5	100	-0.1531	-0.0577	-0.2475	-0.1343
222	11	0.5	67	-0.1594	-0.0295	-0.0631	-0.2128
223	11	0.5	50	-0.1431	-0.0268	0.0126	-0.1283
224	11	0.5	33	-0.0969	0.0088	0.1380	-0.1377
231	11	1.0	100	-0.1608	-0.0627	-0.1717	-0.0663
232	11	1.0	67	-0.1289	-0.0162	-0.1187	-0.0800
233	11	1.0	50	-0.0982	0.0164	-0.0204	-0.1653
234	11	1.0	33	-0.0673	0.0159	0.1146	-0.1197
241	11	2.0	100	-0.0696	0.0494	-0.0918	0.1368
242	11	2.0	67	-0.0548	0.0470	0.0090	0.1099
243	11	2.0	50	-0.1133	0.0139	0.0332	0.0487
244	11	2.0	33	-0.0178	0.0687	0.1741	0.0425
321	9	0.5	100	0.0297	0.0116	-0.2194	-0.0640
322	9	0.5	67	0.0458	0.0096	-0.0240	-0.0446
323	9	0.5	50	0.0597	0.0146	-0.0001	-0.0125
324	9	0.5	33	0.1296	0.0300	0.2075	0.0532
331	9	1.0	100	0.0220	0.0283	-0.1670	0.0819
332	9	1.0	67	0.0278	0.0111	-0.0508	-0.0062
333	9	1.0	50	0.0591	0.0096	0.0173	0.0699
334	9	1.0	33	0.1421	0.0689	0.1851	0.0977
341	9	2.0	100	0.0621	0.0590	-0.0168	0.1534
342	9	2.0	67	0.0649	0.0416	0.0774	0.1466
343	9	2.0	50	0.0770	0.0627	0.1234	0.0993
344	9	2.0	33	0.1440	0.0853	0.1999	0.0494
421	8	0.5	100	0.1815	0.0137	-0.1483	0.0742
422	8	0.5	67	0.1959	0.0646	-0.0592	0.0865
423	8	0.5	50	0.1833	0.0344	0.0351	0.1358
424	8	0.5	33	0.2104	0.0932	0.1509	0.1407
431	8	1.0	100	0.1310	0.0321	-0.1252	0.1119
432	8	1.0	67	0.1795	0.0576	-0.0188	0.0851
433	8	1.0	50	0.1711	0.0565	-0.0226	0.0904
434	8	1.0	33	0.2264	0.0871	0.1940	0.1665
441	8	2.0	100	0.1650	0.0787	0.0399	0.1517
442	8	2.0	67	0.1853	0.0699	0.0710	0.1605
443	8	2.0	50	0.1756	0.0778	0.1540	0.1436
444	8	2.0	33	0.2367	0.1152	0.2098	0.1357

Table 4

Table 5 shows the correlation coefficients between the model and the data for each of the 34 conditions: conditions 1-7 represent test sentences 1-7, and conditions 8-34 represent passes 1-2, or 1-5, for subjects 1-9, as indicated at the left of the table. The square of the correlation coefficient indicates the proportion of variance accounted for by the model, for the data of individual subjects. These figures range between about 36% for subject 2, pass 2, to 90% for subject 1, pass 3.

The positions of the stimuli in the group space, and the weights, are plotted in Figures 7-12. Since the solution is in four dimensions, Figures 7 and 8 show the stimuli and the weights projected on the Dimension-1 vs Dimension-2 plane, and Figures 9 and 10, and 11 and 12, present corresponding plots for the projections on Dim-1 vs Dim-3, and Dim-1 vs Dim-4 planes, respectively. Each weight, and each system, is labelled on the plot by a single digit or letter, and the labels are identified in the table at the bottom of each figure.

The dimensions of the group space were identified with the aid of two sorts of information. First, lines were drawn joining all quartets of systems that differed only in the number of poles. Similar plots were made for quantization step size, and for frame rate. In each case, there was a strong tendency for these lines to be parallel to one or another of the axes: the lines joining systems differing only in number of poles were parallel to Dimension 1; those for frame rate were parallel to

FACTORIAL 444 VOCODER RAW PROXIMITIES, 7 SENT + 27 PASSES, 50 VOCODERS
CORRELATION BETWEEN COMPUTED SCORES AND ORIGINAL DATA FOR SUBJECTS

SENTENCE 1	0.928700
SENTENCE 2	0.842659
SENTENCE 3	0.849738
SENTENCE 4	0.930055
SENTENCE 5	0.934567
SENTENCE 6	0.962202
SENTENCE 7	0.812417
PASS 1, SUBJ 1	0.890779
PASS 1, SUBJ 2	0.708264
PASS 1, SUBJ 3	0.894595
PASS 1, SUBJ 4	0.773677
PASS 1, SUBJ 5	0.876932
PASS 1, SUBJ 6	0.766726
PASS 1, SUBJ 7	0.772804
PASS 1, SUBJ 8	0.898190
PASS 1, SUBJ 9	0.881977
PASS 2, SUBJ 1	0.923826
PASS 2, SUBJ 2	0.598275
PASS 2, SUBJ 3	0.901784
PASS 2, SUBJ 4	0.876386
PASS 2, SUBJ 5	0.860676
PASS 2, SUBJ 6	0.821485
PASS 2, SUBJ 7	0.704472
PASS 2, SUBJ 8	0.911210
PASS 2, SUBJ 9	0.874383
PASS 3, SUBJ 1	0.946425
PASS 3, SUBJ 5	0.873484
PASS 3, SUBJ 6	0.822834
PASS 4, SUBJ 1	0.955949
PASS 4, SUBJ 5	0.900759
PASS 4, SUBJ 6	0.825411
PASS 5, SUBJ 1	0.929301
PASS 5, SUBJ 5	0.871274
PASS 5, SUBJ 6	0.806917

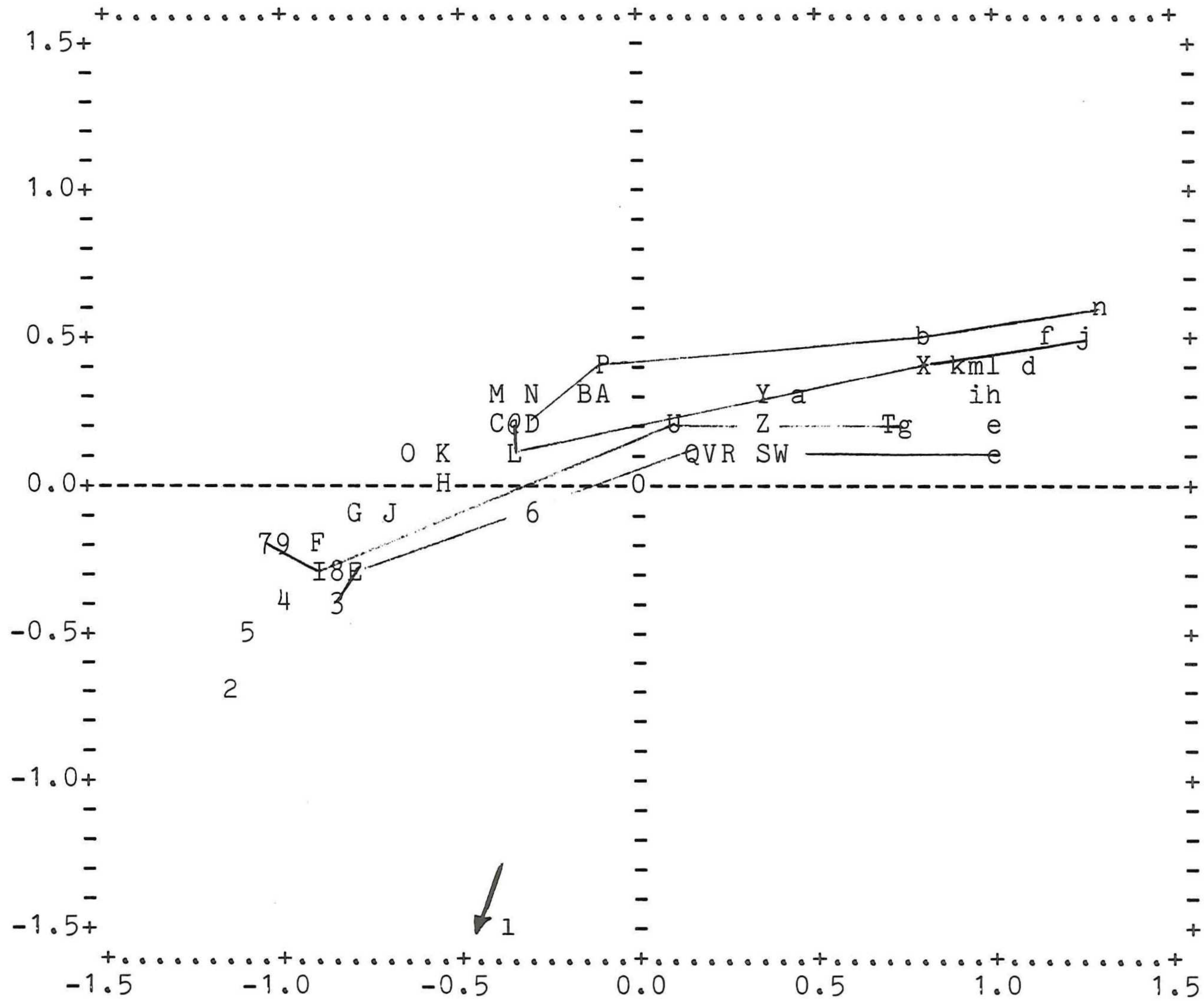
HISTORY OF COMPUTATION
(FINAL ITERATIONS AFTER SETTING MATRIX 2 = MATRIX 3)

ITERATION	CORRELATIONS BETWEEN Y(DATA) AND YHAT(MODEL)	NORMALIZED RESIDUAL VARIANCE (R**2)	(1-R**2)
0	0.860321	0.740152	0.259848
1	0.860355	0.740211	0.259789

Table 5

INSCAL 4-D SOLUTION OF DATA FILE "RX346"

DIM 1 VS. DIM 2 FOR A-MATRIX 2 (VOCODER SYSTEMS)



SCALE FACTOR: MULTIPLY PLOTTED VALUES BY 0.18

SYSTEM KEY: POINT-IDENTIFIER = PQR

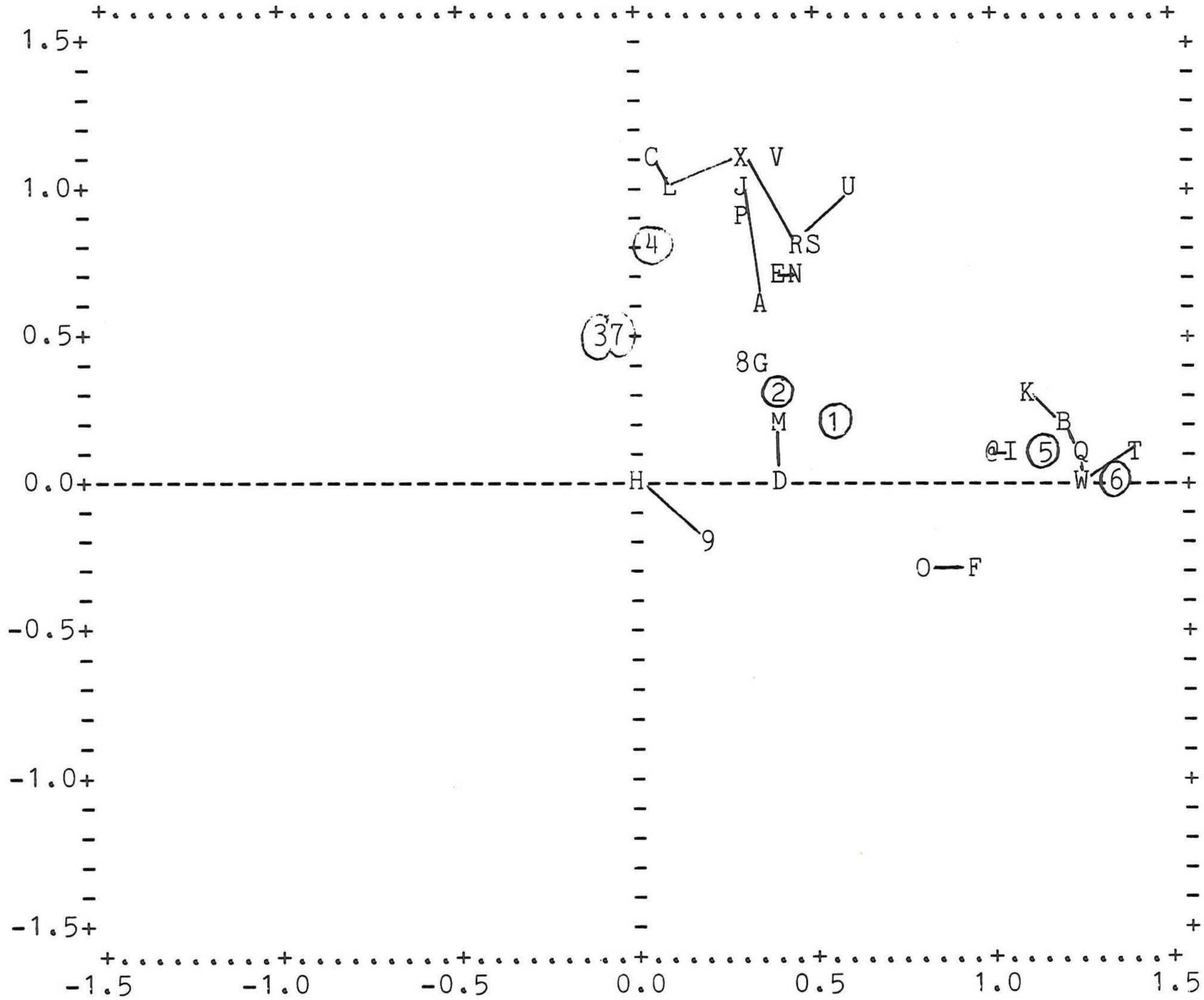
1=000 2=111

3=121	4=122	5=123	6=124	7=131	8=132	9=133	@=134	A=141	B=142	C=143	D=144
E=221	F=222	G=223	H=224	I=231	J=232	K=233	L=234	M=241	N=242	O=243	P=244
Q=321	R=322	S=323	T=324	U=331	V=332	W=333	X=334	Y=341	Z=342	a=343	b=344
c=421	d=422	e=423	f=424	g=431	h=432	i=433	j=434	k=441	l=442	m=443	n=444

Figure 7

INSCAL 4-D SOLUTION OF DATA FILE "RX346"

DIM 1 VS. DIM 2 FOR A-MATRIX 1 (Sentence and Pass WEIGHTS)



SCALE FACTOR: MULTIPLY PLOTTED VALUES BY 0.67

SENTENCE AND PASS KEY

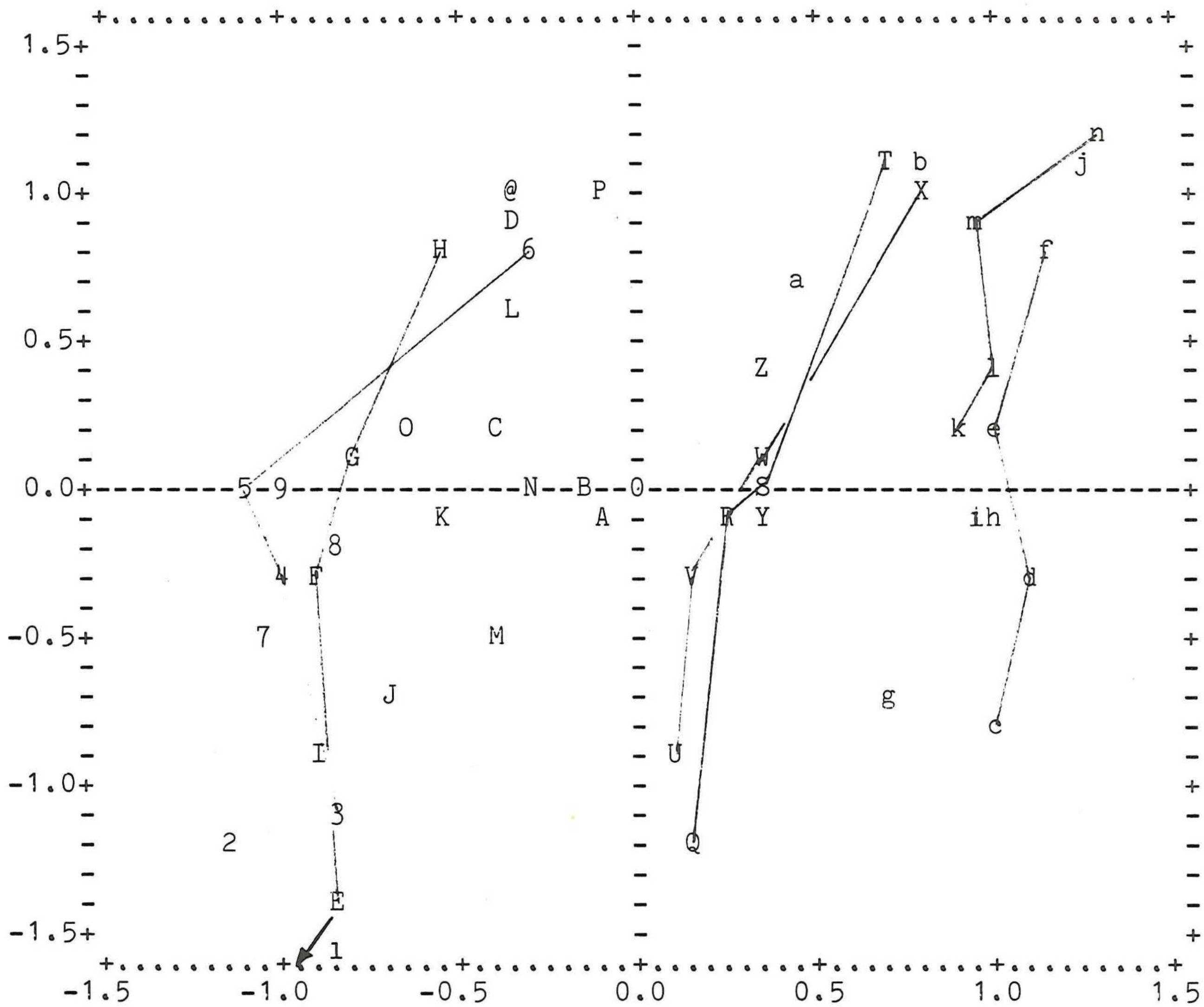
SENT 1=1 2=2 3=3 4=4 5=5 6=6 7=7

SUBJECT	1	2	3	4	5	6	7	8	9
PASS 1:	8	9	@	A	B	C	D	E	F
PASS 2:	G	H	I	J	K	L	M	N	O
PASS 3:	P				Q	R			
PASS 4:	S				T	U			
PASS 5:	V				W	X			

Figure 8

INSCAL 4-D SOLUTION OF DATA FILE "RX346"

DIM 1 VS. DIM 3 FOR A-MATRIX 2 (VOCODER SYSTEMS)



SCALE FACTOR: MULTIPLY PLOTTED VALUES BY 0.18

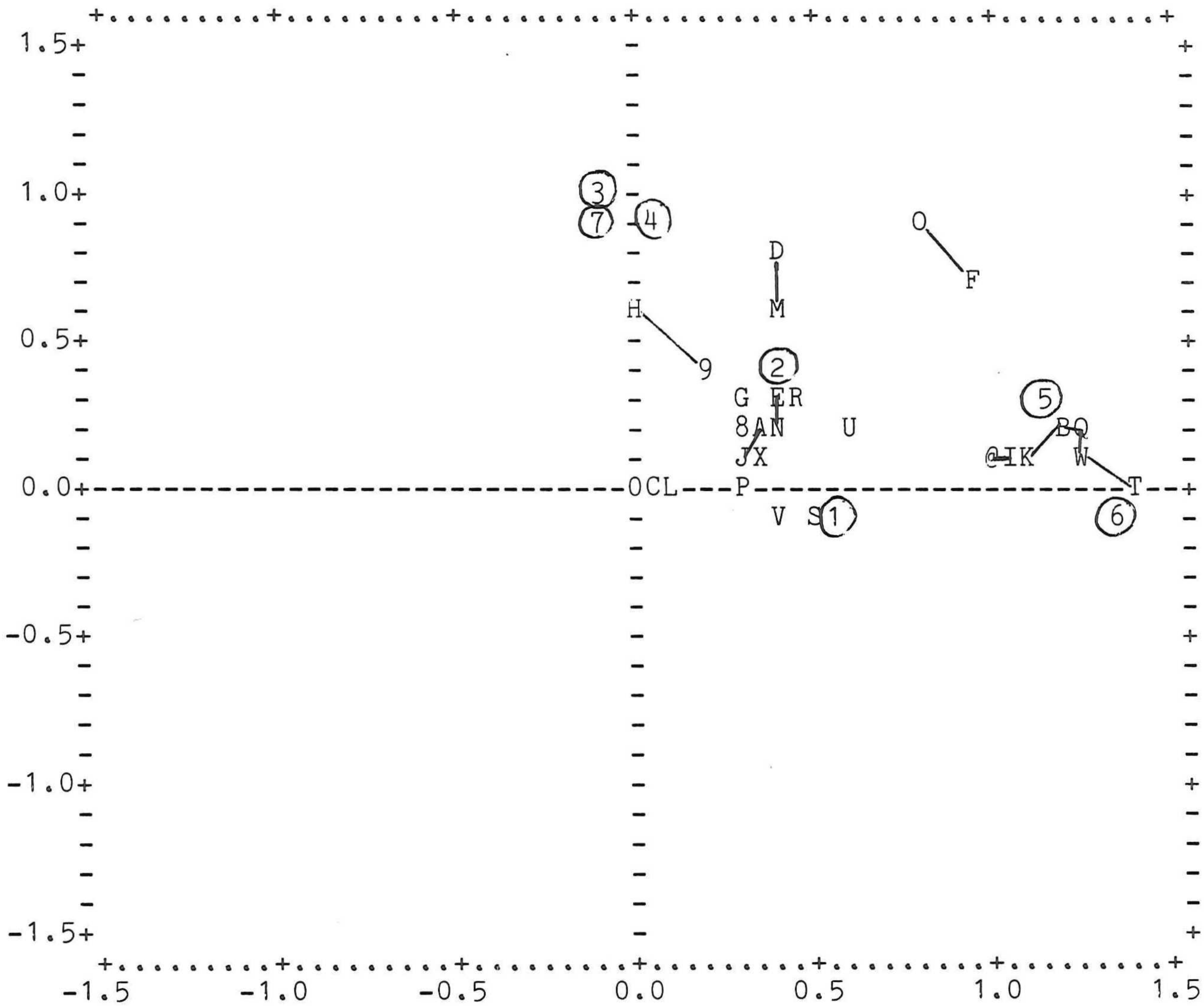
SYSTEM KEY: POINT-IDENTIFIER = PQR

1=000 2=111
3=121 4=122 5=123 6=124 7=131 8=132 9=133 @=134 A=141 B=142 C=143 D=144
E=221 F=222 G=223 H=224 I=231 J=232 K=233 L=234 M=241 N=242 O=243 P=244
Q=321 R=322 S=323 T=324 U=331 V=332 W=333 X=334 Y=341 Z=342 a=343 b=344
c=421 d=422 e=423 f=424 g=431 h=432 i=433 j=434 k=441 l=442 m=443 n=444

Figure 9

INSCAL 4-D SOLUTION OF DATA FILE "RX346"

DIM 1 VS. DIM 3 FOR A-MATRIX 1 (Sentence and Pass WEIGHTS)



SCALE FACTOR: MULTIPLY PLOTTED VALUES BY 0.67

SENTENCE AND PASS KEY

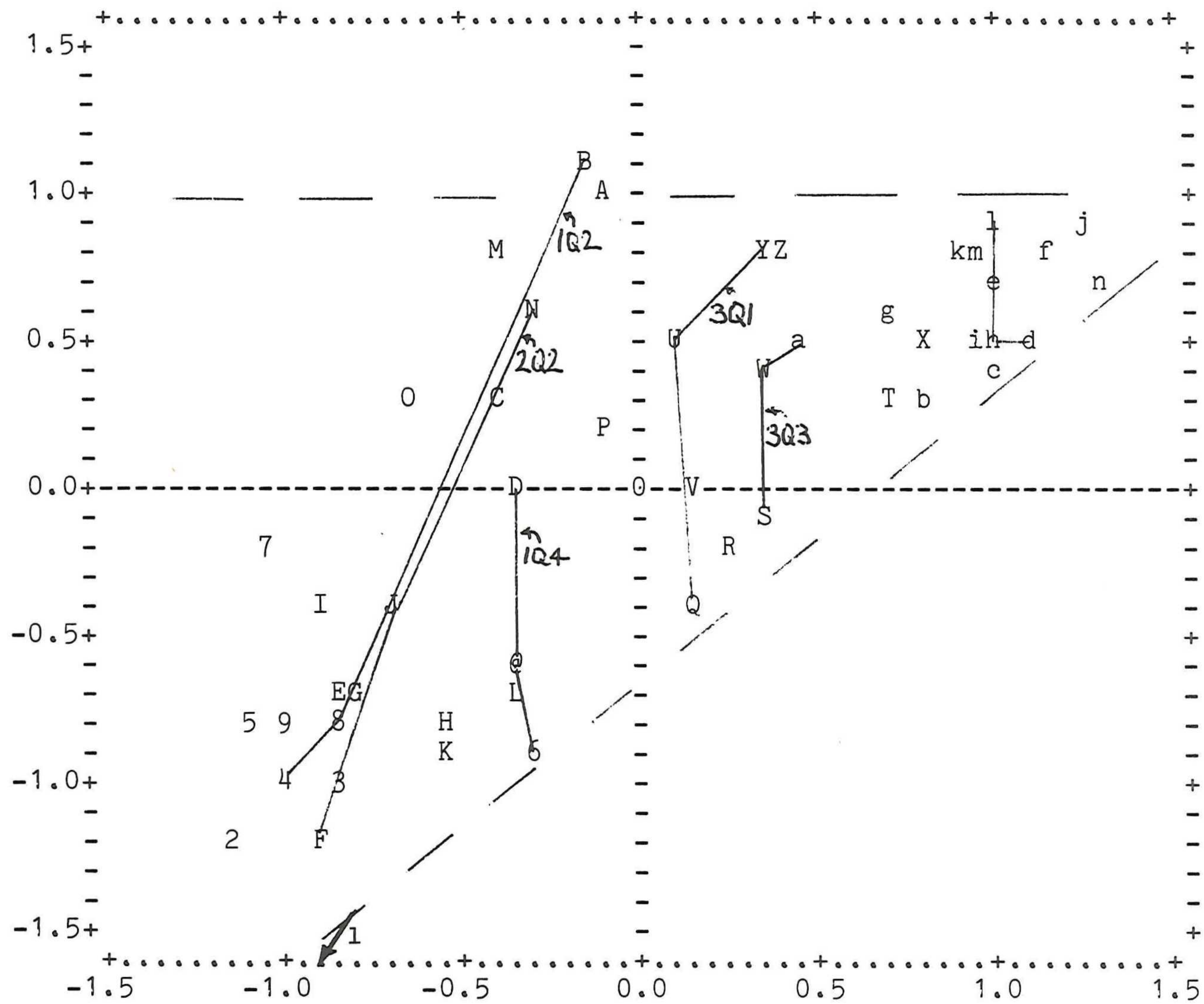
SENT 1=1 2=2 3=3 4=4 5=5 6=6 7=7

SUBJECT	1	2	3	4	5	6	7	8	9
PASS 1:	8	9	@	A	B	C	D	E	F
PASS 2:	G	H	I	J	K	L	M	N	O
PASS 3:	P				Q	R			
PASS 4:	S				T	U			
PASS 5:	V				W	X			

Figure 10

INSCAL 4-D SOLUTION OF DATA FILE "RX346"

DIM 1 VS. DIM 4 FOR A-MATRIX 2 (VOCODER SYSTEMS)



SCALE FACTOR: MULTIPLY PLOTTED VALUES BY 0.18

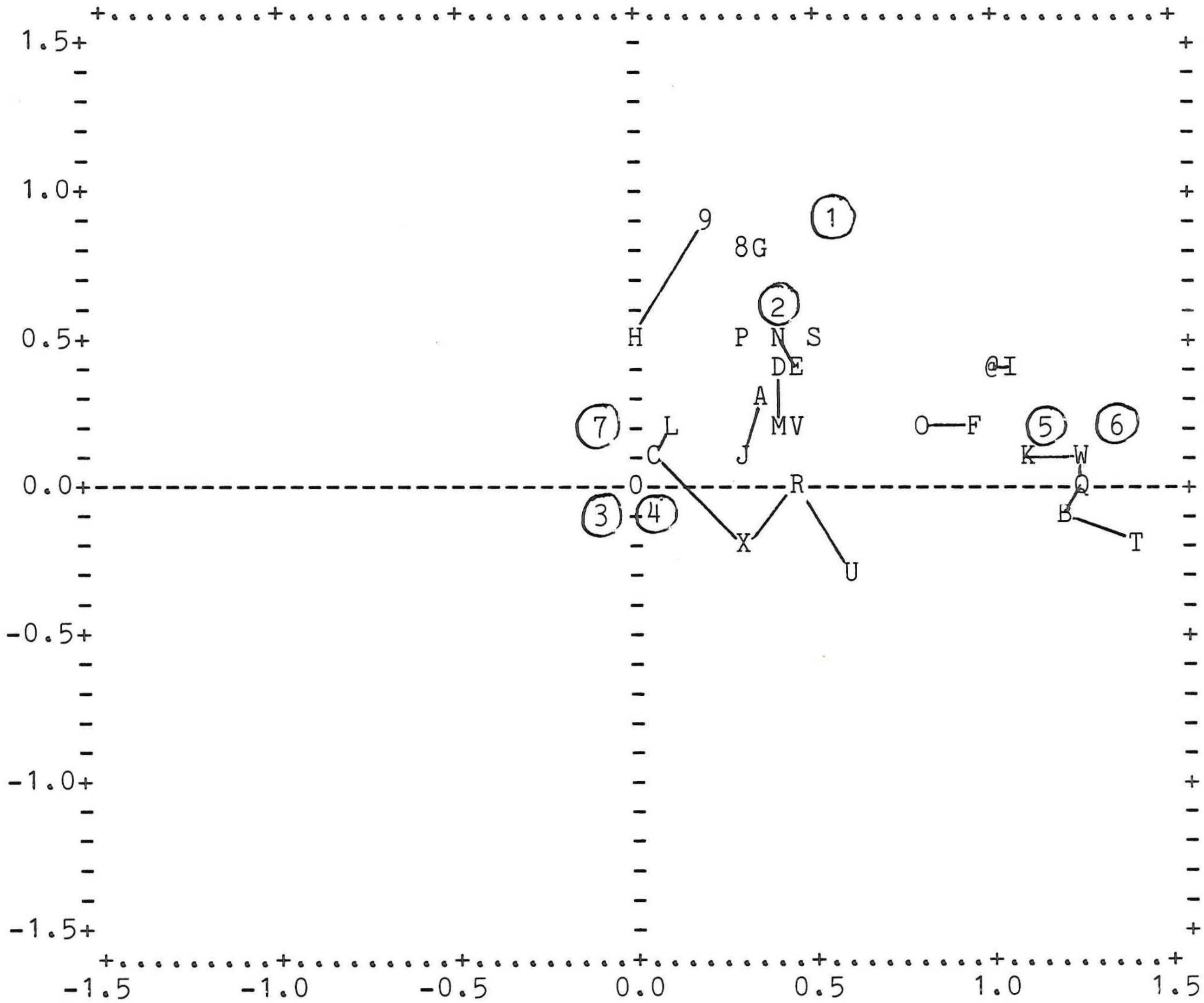
SYSTEM KEY: POINT-IDENTIFIER = PQR

1=000 2=111
3=121 4=122 5=123 6=124 7=131 8=132 9=133 @=134 A=141 B=142 C=143 D=144
E=221 F=222 G=223 H=224 I=231 J=232 K=233 L=234 M=241 N=242 O=243 P=244
Q=321 R=322 S=323 T=324 U=331 V=332 W=333 X=334 Y=341 Z=342 a=343 b=344
c=421 d=422 e=423 f=424 g=431 h=432 i=433 j=434 k=441 l=442 m=443 n=444

Figure 11

INSCAL 4-D SOLUTION OF DATA FILE "RX346"

DIM 1 VS. DIM 4 FOR A-MATRIX 1 (Sentence and Pass WEIGHTS)



SCALE FACTOR: MULTIPLY PLOTTED VALUES BY 0.67

SENTENCE AND PASS KEY

SENT	1=1	2=2	3=3	4=4	5=5	6=6	7=7

SUBJECT	1	2	3	4	5	6	7	8	9
PASS 1:	8	9	@	A	B	C	D	E	F
PASS 2:	G	H	I	J	K	L	M	N	O
PASS 3:	P				Q	R			
PASS 4:	S				T	U			
PASS 5:	V				W	X			

Figure 12

Dimension 3; and those for quantization step size were parallel to Dimension 4. The only dimension not identifiable in this way was Dimension 2. A typical sample of these lines appears in Figures 7, 9, and 11. Not all the lines are included, because then the figures become so cluttered that they are hard to read.

The foregoing tentative identification of axes 1, 3, and 4 were confirmed by correlating the system coordinates in the group space with the system parameters. Table 6 shows the correlation matrix for the number of poles, quantization step size, frame rate, bits per frame (labelled B/F in Table 6), the overall bit rate (labelled KBPS), and the four INDSCAL coordinates. System 000 (110 kbps PCM, the "undistorted original") was excluded from these correlations, since it was not possible to specify the number of poles, quantization step size, or frame rate. With 49 systems, a correlation coefficient of 0.45 achieves significance at $P < 0.001$, and one of 0.28 is significant at $P < 0.05$. As expected, Dimension 1 correlated very highly with the number of poles ($R=0.88$), and insignificantly with quantization step size ($R=0.2$) or with frame rate ($R=-0.2$). Similarly, Dimension 3 correlated highly ($R=-0.85$) with frame rate, and insignificantly with number of poles ($R=0.17$). The correlation with quantization step size was also significant ($R=0.37$). Dimension 4 correlated significantly both with number of poles ($R=-0.61$) and with quantization step size ($R=0.61$). This can be explained easily with the aid of Figure 11F, which shows that there was a wide spread of systems along Dimension 4 (ordinate) at the left of the

FACTORIAL 444: CORRELATE P, Q, R, B/F, KBPS VS. INSCL DIMS 1-4

	P	Q	R	B/F	KBPS	DIM 1	DIM 2	DIM 3	DIM 4
P	1.00								
Q	-0.04	1.00							
R	0.04	-0.04	1.00						
B/F	0.58	-0.80	0.09	1.00					
KBPS	0.32	-0.37	0.87	0.54	1.00				
DIM 1	-0.88	0.20	-0.20	-0.65	-0.48	1.00			
DIM 2	-0.64	0.58	-0.35	-0.86	-0.69	0.81	1.00		
DIM 3	-0.17	0.37	-0.85	-0.39	-0.90	0.41	0.61	1.00	
DIM 4	-0.61	0.61	0.01	-0.85	-0.38	0.77	0.83	0.32	1.00

Table 6

figure, where the systems have 13 or 11 poles, but a narrow spread at the right side, where systems have only 9 or 8 poles. Coarsening the quantization step size had little effect on quality when only 8 or 9 poles were used, since the signal was already badly degraded because too few poles had been used for adequate spectral accuracy. On the other hand, where sufficient poles were used, coarsening the quantization had a substantial effect on quality. The same result was observed -- although much less clearly -- in Figure 5 of QPR No. 9, part 2, where there was a large effect on quality of coarsening quantization at high bit rates, which became insignificant at lower bit rates, where the quality had already been lost by using too few poles.

Dimension 2 correlated highly with both P and Q, and less so (but still significantly) with R. It is not possible to identify Dimension 2 without the additional information available from the sentence-weights, which we discuss below.

Several other points should be made about the correlations shown in Table 6. First, as would be expected in a factorial study, P, Q, and R are not significantly correlated with each other. Second, bits per frame (B/F) is highly correlated with poles, and even more highly with quantization, but unrelated to frame rate. This reflects the fact that the range of quantization values used had a larger effect on bits per frame than did the range of number of poles. Overall bit rate, on the other hand, was much more highly affected by frame rate than by

either poles or quantization, or even by bits per frame. In part, this was because frame rate and overall bit rate were linearly related, since pitch and gain were transmitted at the same frame rate as spectral data. The effects on overall rate of changing P and Q, on the other hand, were diluted by the 11 bits of pitch and gain coding transmitted with each frame of spectral data.

From the right hand four columns in Table 6, it can be seen that there are considerable correlations between the four INDSCAL dimensions. All the correlations are significant, most of them highly significant. The fact that the correlations are so high makes it impossible to argue that the perceptual effects of changing the three vocoder parameters were independent, as we had hoped to show. On the other hand, such hopes were perhaps unrealistic. Clearly, there was a large interaction between the quantization step size and the number of poles, as pointed out above. There is likely to be a similar interaction between quantization step size and frame rate. Coarsening the quantization step size reduces the best static spectral accuracy that can be achieved, as we have argued in earlier reports. But reduced static spectral accuracy also sets an upper limit on the best <u>dynamic</u> accuracy that can be achieved, so that

quantization step size is bound to interact with frame rate, as well as with number of poles. The degradation caused by too coarse quantization is likely to be most noticeable when the frame rate is high, since these conditions maximize the chances that successive frames of spectral data will alternate about a quantization step boundary. This effect, also, was suspected from earlier work (see QPR No. 9, part 2, Figure 6, where quality improved as frame rate was reduced, for systems 24R and 14R, which had 11 and 13 poles respectively, and coarse quantization of 2.0 dB).

Now we turn to the second source of information available for identifying the dimensions of the INDSCAL solution, the sentence-weights. The subject-weights are plotted with the sentence-weights in Figures 8, 10, and 12, although they are useful only for looking at within-subject consistency, and for seeing how large were the differences between subjects in the perceptual salience of the different dimensions. In the figures, sentences weights are ringed, to accentuate them, and the weights for individual subjects are joined by lines, where this is possible without cluttering the figure.

Weights on Dimension 1: Earlier results had suggested that there was a strong interaction between the fundamental frequency of the talker, and the amount of degradation of speech quality that resulted from reducing the number of poles (see QPR No. 9, part 2, p 33). This result is confirmed in the INDSCAL solution

by the ordering of the sentence-weights on Dimension 1 (see Figure 8). The sentence-weights are ordered (with one inversion) by the average fundamental frequency of the speaker. The sentence with the lowest fundamental (sentence 6, DK6, F0 = 97 Hz) had a large positive weight; Sentences 5 (JB5), 1 (JB1), and 2 (DD2), had progressively smaller weights, and (except for 1 and 5) progressively higher fundamental; and the three female sentences (3, 4, and 7) have zero, or even slightly negative weights. (We will return to the question of negative weights below.)

<u>Weights on Dimension 2</u>: The arguments for interpreting Dimension 2 as related to the sex of the speaker, or the breathiness of the speakers voice which tends to be more pronounced in females, are less strong than those for the other dimensions. The three female sentences (Nos 3, 4, and 7) had virtually zero weights on Dimension 1, which has been identified with the effect of poles on male voices. Earlier plots of mean degradation rating against overall bit rate, performed separately for each sentence, showed that the effects of poles on female voices were both smaller and different from the effects on male voices. Therefore, Dimension 2 was needed to model the differences between male and female voices. This fits in with the observation that LPC modelling of the spectra of female voices is far less satisfactory than for male voices. The high correlations between system coordinates on Dimension 2 and poles and quantization probably reflect the less adequate spectral

modelling of female voices. Further, the pulse excitation used in present vocoders does not generate good voice quality for females, perhaps because of differences in laryngeal dynamics. In this regard, it is interesting that the sentence with the largest weight on Dimension 2 (sentence 4, AR4) was also much the most "breathy." Secondly, the male voice with the largest weight on Dimension 2 (sentence 2, DD2) was the breathiest of the male talkers.

Weights on Dimension 3: The sentence-weights on Dimension 3 suggest that frame rate may have influenced quality more through the parameters of pitch and gain than through spectral variables. On the basis of spectral variables only, the fastest-moving sentences (Nos 4 and 3) should have the largest weights, followed by the general sentences (Nos 5, 6, and 7), followed by the slow-moving sentences (Nos 2 and 1). Although the rank order of the observed sentence weights correlates significantly with this predicted order (Spearman $r=0.77$, $P < 0.05$), there is a major discrepancy with respect to sentences 6 and 7, which are the same sentence spoken by a male and female speaker, respectively. Sentence 7 has a high weight, and sentence 6 has a low weight on Dimension 3. A second variable that might correlate with the observed weights is the range of fundamental frequency (F0) in the sentence. Earlier work showed this to be correlated with average frame rate, in a VFR system, suggesting that sentences with very variable F0 might require a higher frame rate than needed for purely spectral purposes, to represent the pitch

contour adequately. Since pitch and gain were transmitted at the same frame rate as spectral data in the present experiment, inadequate frame rate might degrade quality either because spectral data was not updated fast enough, or because pitch and gain data were not updated fast enough. In support of this argument, the rank order of the sentences by relative standard deviation of F0 (s.d. / mean) was significantly correlated with the observed weights on Dimension 3 (Spearman $r=0.84$, $P < 0.05$).

Weights on Dimension 4: The weights for Dimension 4 (Figure 12) support the identification of Dimension 4 as representing the effects of quantization step size. We argued in QPR No. 1 that too coarse quantization should show up most strongly in sentences that are all voiced, and contain only relatively slow spectral changes -- that is, Sentence 1 ("Why were you away a year, Roy?") and Sentence 2 ("Nanny may know my meaning."). This expectation was confirmed by the fact that these two sentences were the only two that had significant weights on Dimension 4, which we have already identified with quantization step size. Further, the remaining sentence weights are ordered on Dimension 4 as would be predicted by this argument: following the highly weighted "slow-moving" sentences are the three general sentences, Nos 5, 6, and 7, and following these are the fast-moving sentences, Nos 3 and 4, which also contain much frication energy, in which quantization step size should have little, if any, effect.

Negative weights: The incidence of negative weights can be taken as an indication of how well a set of data match the assumptions made by the INDSCAL model. Twenty out of the 136 weights shown in Table 3 are negative, but all except two or three are small enough to be ignored. The two largest negative weights were both related to a single subject (No 9) on Dimension 2. We do not regard this rate of incidence as damaging to our use of INDSCAL. Several options are open, should the incidence be larger in later applications of the method. One possibility is to use the need for negative weights in modelling a particular subject's data as a criterion for discarding that subject's data, and repeating the analysis without it. Secondly, the fact that negative weights sometimes occur can be taken as a criticism of the way the INDSCAL algorithm was implemented by Bell Telephone Laboratories, from whom we acquired the program. It would be possible to rewrite the least squares procedure (in seven dimensions!) so as to incorporate the restriction that weights must be non-negative, by using a constrained optimization procedure but we do not consider this effort worthwhile, at least at present. A further possibility would be to use a more recent alternating least squares algorithm that includes INDSCAL as a special case. This program, called ALSCAL, was described in a recent paper [7].

V. REFERENCES

1. R. Viswanathan, J. Makhoul and R. Wicke, "The Application of a Functional Perceptual Model of Speech to Variable-Rate LPC Systems," Proc. 1977 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 219-222, May 1977.

2. A. W. F. Huggins, R. Viswanathan and J. Makhoul, "Quality Ratings of LPC Vocoders: Effects of Number of Poles, Quantization and Frame Rate," Proc. 1977 IEEE International Conference ASSP, pp. 413-416, May 1977.

3. BBN Quarterly Progress Report on Command and Control Related Computer Technology, BBN Report No. 3520, March 1977.

4. R. Viswanathan and J. Makhoul, "Specifications for ARPA-LPC System II," NSC Note 82, Feb. 1976.

5. BBN Quarterly Progress Report on Command and Control Related Computer Technology, BBN Report No. 3093, June 1975.

6. J. D. Carroll, "Individual differences and multidimensional scaling," in R. N. Shepard, A. K. Romney, & S. Nerlove (Eds) <u>Multidimensional scaling: Theory and applications in the behavioral sciences</u>, Vol. 1, New York: Seminar Press, 1972, pp. 105-155.

7. Y. Takane, F. W. Young, and J. de Leeuw "Nonmetric individual differences multidimensional scaling: an

alternating least squares method with optimal scaling features," Psychometrica, Vol. 42, pp. 7-68, 1977.

8. M. Wish and J. D. Carroll "Applications of individual differences scaling to studies of human perception and judgment," in E. C. Carterette and M. P. Friedman (Eds) Handbook of Perception, Volume 2: Psychophysical judgment and measurement. New York: Academic Press, 1974, pp. 449-491.